Nonrigid Surface Tracking, Analysis and Evaluation

Li, Wenbin

Award date:
2014

Awarding institution:
University of Bath

Nonrigid Surface Tracking, Analysis and Evaluation

submitted by

Wenbin Li

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Computer Sciences

October 2013

Abstract

Estimating the dense image motion or optical flow on a real-world nonrigid surface is a fundamental research issue in computer vision, and is applicable to a wide range of fields, including medical imaging, computer animation and robotics. However, nonrigid surface tracking is a difficult challenge because complex nonrigid deformation, accompanied by image blur and natural noise, may lead to severe intensity changes to pixels through an image sequence. This violates the basic intensity constancy assumption of most visual tracking methods. In this thesis, we show that local geometric constraints and long term feature matching techniques can improve local motion preservation, and reduce error accumulation in optical flow estimation. We also demonstrate that combining RGB data with additional information from other sensing channels, can improve tracking performance in blurry scenes as well as allow us to create nonrigid ground truth from real world scenes.

First, we introduce a local motion constraint based on a laplacian mesh representation of nonrigid surfaces. This additional constraint term encourages local smoothness whilst simultaneously preserving nonrigid deformation. The results show that our method outperforms most global constraint based models on several popular benchmarks. Second, we observe that the inter-frame blur in general video sequences is near linear, and can be roughly represented by 3D camera motion. To recover dense correspondences from a blurred scene, we therefore design a mechanical device to track camera motion and formulate this as a directional constraint into the optical flow framework. This improves optical flow in blurred scenes. Third, inspired by recent developments in long term feature matching, we introduce an optimisation framework for dense long term tracking – applicable to any existing optical flow method – using anchor patches. Finally, we observe that traditional nonrigid surface analysis suffers from a lack of suitable ground truth datasets given real-world noise and long image sequences. To address this, we construct a new ground truth by simultaneously capturing both normal RGB and near-infrared images. The latter spectrum contains dense markers, visible only in the infrared, and represents ground truth positions. Our benchmark contains many real-world scenes and properties absent in existing ground truth datasets.

Acknowledgements

This thesis would not be possible without the supervision, support and help of so many people around during my PhD study.

My first and sincere appreciation goes to my supervisor, Dr. Darren Cosker. In his wisdom and infinite patience, Darren made a great effort to supervise me during the last three years, from theoretical creations to the wording in professional publications. He would be always a container for countless brilliant ideas. His words of encouragements and supervision have helped me overcome obstacles and construct my knowledge base in fields. I have learned from him not only how to conduct achievements in research, but also how to be a creative pioneer in the industry.

I also want to thank my thesis examiners, Prof. Brian Wyvill and Dr. Lourdes Agapito. Having been TA for Brain's first Graphics unit at Bath, I always benefited from his deep insight in research and teaching, as well as his humour and wisdom for life. The pioneering contribution of Lourdes in nonrigid structure from motion has inspired our major work in this thesis. She generously share their nonrigid tracking benchmark and critical experience, which motivates me to improve our nonrigid optical flow algorithm.

Thanks to other academics and researchers with whom I worked together during my thesis: Dr. Matthew Brown, Dr. Chuan (Chris) Li, Dr. Yi-Zhe Song, Rui Tang and Dr. Leon Watts. Matt has generously shared his perspective towards computer vision research as well as his fantastic spectral material to allow me to construct my own experience in multispectral imaging. He also tolerated my terrible writing and gave me many insightful comments on my first CVPR paper. Chris is personally a great mentor and friend to me. He helped me generously to pick up the optical flow formula. His attitude towards research, patience and concentration, always encouraged me especially when I was dejected and frustrated. Song was my first teacher for MATLAB and Unix stuffs, simply being a big brother and always there. Rui shared his great insight views in the facial stuffs and so much fun in both research and life. I really appreciated the words Leon shared with me when my paper was rejected again in ECCV: "The work got rejected but doesn't mean it's useless but just not perfect right now."

I feel very privileged to share the open office with all the MTRC members and other colleagues, David Pickup, Dr. Hongping Cai, Dr. Lizzy Gabe-Thomas, Andrew Chinery, Tom Saunders, Qi Wu, Gang (Garry) Ren, Bidan Huang, Nick Westlake,

Han Gong, JeeHang Lee, Shadi Basurra, Saeid Ardakani. We shared so much fun and laughter together within such an open office. I particularly thank to Garry who extended my view into the HCI and the digital entertainment. We have spent great time together to discuss some marvellous and delicate devices and ideas from such a digital world.

I want to thank the department administration, Dilly Brownlow and Susan Paddock. Both of them were kindly receiving tens and hundreds of parcels for me in the last three years. I appreciated the attendance of all the students who have chosen the units I lectured or tutored.

Finally, my deepest gratitude goes to my parents and my wife, Tingting Li, for their long-time support and love, which is beyond words and tones. This thesis is particularly dedicated to my grandfather and father in law. My grandfather is known as one of the best architects and structural engineers in China in the last four decades. With his broad knowledge in mathematics and engineering, he helped me to construct the mathematical skills and eventually led me to research in science. This thesis is also impossible without the support of my father in law. I have learned from him how to be a man of integrity, humility, patience and kindness. I deeply appreciated one of his famous sayings "working hard and never give up" which motivated me to conquer many obstacles during my low life in the research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are living in a dynamic world and thus surrounded by the perceptual motion that is often observed on object surfaces. Most of the perceptual motion is not ideally rigid with low degree of freedom but often behaves interactively free-form and locally complex, namely *Nonrigid Motion*. Similar to the visual texture and surface geometry, such nonrigid motion provides a rich set of information for humans to understand the surrounding environment and perform the interaction. For instance, humans can interpret the emotions of other people by reading the motion cues on a human face, which is the most typical nonrigid surface in the real-world scene. While humans can easily identify a deformable human face, the computer, however, lacks such capability of tracking nonrigid surface rendered by a great amount of pixels from a 2D image sequence. This difficulty is addressed by a *Dense Tracking* process in the computer vision community.

Dense tracking e.g. optical flow estimation, is identified as a solution to locating most of pixels through multiple images, which commonly follows a fundamental *Brightness Constancy* assumption that the pixel brightness remains unchanged. This technique underpins the research in many other subareas of higher level computer vision such as video augmentation, motion capture and visual effects. Tracking a nonrigid surface in a real-world scene is a difficult task. The main challenge lies in the temporal violation of the pixel brightness constancy, which is caused by the complex nature of nonrigid motion, as well as the accompanying difficulties during images capture i.e. camera shake, repeated texture, arbitrary occlusions and large displacements. Such violation of brightness constancy constraint leads to unpredictable errors during nonrigid surface tracking.

Violation of *Brightness Constancy* is a common issue, which mathematically yields a smaller number of equations than unknowns. Typical solvers are normally categorised in two ways: the first strategy lies on the acquisition of more known variables from extra input data, in particular additional images or reliable landmarks. However such extra

Figure 1-1: Nonrigid surface tracking using our approach (Chapter 3). The test sequence is from [108].

information is often manually selected and difficult to obtain in practice. The other strategy is to impose additional *Constraints* on the problem, i.e. further assumptions on texture or motion behaviour derived beforehand. In mathematical terms, a constraint may represent the geometric relation, physical model or statistical assumptions, which gives significant influence on the quality and efficiency of the host algorithm performance. Hence, the use of constraint the results in requiring less input data and is often more practical for real-world sequences.

In Chapter 3, we present the initial focus on a local motion constraint, namely *Laplacian Mesh* constraint, to improve pairwise optical flow estimation on a typical nonrigid surface i.e. *cloth*. The *Laplacian Mesh* constraint is presented as the inherent geometric relation between a pixel and its adjacent neighbours: the movement of connected vertices (pixels) on a deformable surface behaves similarly within a small *neighbourhood* even when some vertices are occluded. This observation also holds in each pixel within a real-world nonrigid surface. Combining this constraint and a variational optical flow framework, we obtain highly accurate correspondence between an image pair containing nonrigidly deforming objects. Our experiments demonstrate the success and outperforms many previous methods on several benchmark datasets (See Fig. 1-1[1]).

In the real-world photography, the blur caused by camera shake often obscures image properties, which thus gives rise to brightness consistency deterioration. Our second contribution detailed in Chapter 4, is an extension to optical flow algorithms against such artefacts. We exploit that the inter-frame blur in video sequence with a standard frame rate (e.g. 24 FPS) is near linear. Such blur can therefore be roughly derived by the camera motion. This discovery suggests an additional information channel to the conventional camera. We design an imaging system by attaching a 3D Pose&Position Tracker to an ordinary camera in order to obtain such camera motion trajectory. The

[1]This thesis gives best view of all the images involved in the electronic version.

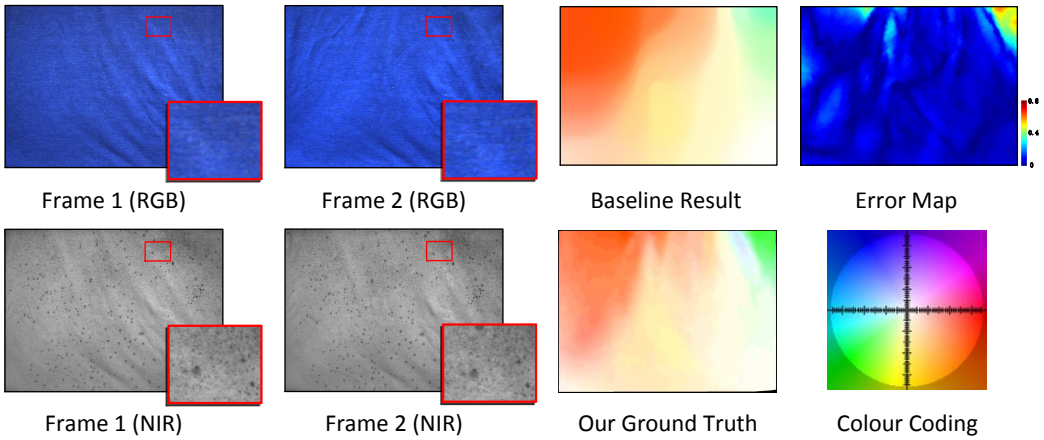


Figure 1-2: An example sequence *featureless* from our nonrigid ground truth dataset, highlighting the dense invisible NIR patches in a large textureless region of a nonrigid surface.

camera motion information is then applied as a directional constraint to enhance the optical flow estimation in a blurred scene. We find this directional constraint efficient, as well as adaptable as a filter which is superior to other existing techniques including high quality image deblurring. Hence, we advance the state-of-the-art in a broader area of dense tracking with motion deblurring.

Although our work significantly reduces the error in pairwise tracking, small errors may still exist. Such errors can be accumulated between frames over time, which leads to deviation from the correct tracking trajectory in a long image sequence. This is the well-known *Drift* problem in long term tracking. Common solutions include the use of learning based prior or reliable landmarks. However, such solvers are resource intensive and dependent on the quantity of training data and manual intervention. In Chapter 5, we introduce a feature-based automatic scheme to detect reliable matching patches and frames within a long image sequence. Such reliable patches and frames have direct correspondence to the reference. These can significantly reduce drift by means of shortening the tracking distances for local regions throughout the entire sequence, as well as booming the computational speed by enabling tracking in parallel.

Our work on nonrigid surface is not limited to dense tracking. The quantitative evaluation of tracking algorithms is challenging particularly given long real-world scenes and nonrigidly deformable surfaces. An existing strategy to capture *Ground Truth* correspondence from real-world scenes is to use the *Stop-Motion* scheme [5]: a scene is first captured under normal lighting; and then objects are frozen to capture a feature-rich image of the same scene under ultraviolet lighting. Such a capture scheme enables real-world ground truth capture but is limited by the lack of motion blur and the difficultly in capturing long image sequences. In Chapter 6, we discuss nonrigid ground truth construction using *RGB&Near-Infrared Imaging* and *Infrared Visible Dyes*. We simultaneously capture both normal RGB and Near-Infrared images. The latter con-

tains dense markers – visible only in an infrared spectrum – representing the ground truth positions. Our system produces nonrigid ground truth over long video sequences and preserves realistic photometric effects (See Fig. 1-2). This may also be adopted to capture other types of deformable objects, thus opening ground truth acquisition opportunities in other difficult-to-track problems.

1.1 Main Contributions

In summary, this thesis contains four major contributions as follows:

- In Chapter 3 we propose a novel Laplacian mesh constraint and apply it to pairwise optical flow estimation for nonrigid surfaces.
- In Chapter 4 we introduce a sensor-aided motion constraint and examine the use in pairwise optical flow estimation for blurred scenes due to camera shake.
- In Chapter 5 we discuss *Drift* reduction in long image sequences by combining long term feature identification with pairwise optical flow estimation.
- In Chapter 6 we capture a new nonrigid ground truth benchmark using hidden features in a near-infrared spectrum.

The rest of this thesis is organised as follows: in Chapter 2 we give a comprehensive study on pairwise optical flow estimation, as well as introduction to the background of other concepts involved i.e. nonrigid surface representation, camera motion deblurring and infrared imaging. Then we conclude in Chapter 7.

1.2 Related Publications

The following publications related to this work were produced during my PhD research:

[77] **W. Li**, D. Cosker, M. Brown, and R. Tang, *Optical Flow Estimation using Laplacian Mesh Energy*, in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), IEEE, June 2013, pp. 2435–2442.

[75] **W. Li**, Y. Chen, J. Lee, G. Ren, and D. Cosker, *Robust Optical Flow Estimation for Continuous Blurred Scenes using RGB-Motion Imaging and Directional Filtering*, in Proceeding of IEEE Winter Conference on Application of Computer Vision (WACV'14), 2014. (awarded as **Best Student Paper**)

[76] **W. Li**, D. Cosker, and M. Brown, *An Anchor Patch Based Optimisation Framework for Reducing Optical Flow Drift in Long Image Sequences*, in Proceeding of Asian Conference on Computer Vision (ACCV'12), Springer, November 2012, pp. 112–125.

In addition, portions of the work described in this thesis were included in the following non-refereed materials:

W. Li, D. Cosker, and M. Brown, *A Nonrigid Ground Truth Dataset and Multispectral Optical Flow Estimation using Combined RGB and Near-Infrared Imaging*, MTRC Technical Report, University of Bath, March 2013, pp. 1–8.

W. Li and D. Cosker, *Video Image Registration using A Concatenative Approach*, Poster presentation at AVA/BMVA Spring (AGM) Meeting, April 2011.

[135] R. Tang, D. Cosker, and **W. Li**, *Global Alignment for Dynamic 3D Morphable Model Construction*, in Proceeding of Workshop on Vision and Language (V&LW'12), 2012.

Chapter 2

Background

In this chapter, we first review pairwise optical flow estimation e.g. Horn and Schunck's, and Brox's variational frameworks, as well as the effects of common difficulties (motion boundaries, occlusions and blur) and nonrigid deformation. We then discuss related work of the Laplacian representation for deformable surfaces and image restoration from a blurry scene. We also move into the near-infrared imaging area to expose its potential in visible image enhancement. Finally, we outline the nonrigid tracking challenges this thesis addresses.

2.1 Optical Flow Estimation

In the last two decades, optical flow estimation has been considered to be one of the fundamental research topics by the computer vision community. Optical flow is defined as *apparent motion* [56] of brightness patterns or image properties. In practice, optical flow is also formulated as visible pixel displacements between two consecutive images. However, optical flow is not always the same as *motion field* that is known as the 2D projection of the 3D motion in the world coordinate. Because *motion field* includes the motion of occluded pixels which is often absence in optical flow. In this thesis, we consider robust optical flow estimation – considering both *apparent motion* and *motion field* estimation. Such optical flow (dense correspondences) is frequently involved in the other high level vision and/or graphics areas such as segmentation [160] scene understanding [52] and highly detailed animation [8, 16].

In this section, we consider the goal of the optical flow estimation, which is to compute the optical flow field $\mathbf{w}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^T$ between two images I_t and I_{t+1} : $(\Omega \subset \mathbb{R}^3) \rightarrow \mathbb{R}$ with time axis t where $\mathbf{x} = (x, y)^T$ denotes a pixel location in the spatial domain of an image. In this case, the fundamental assumption for the optical flow estimation is the *Brightness Constancy* where the pixel brightness or image property is assumed not to change through the entire image sequence. Such an assumption can

be mathematically represented as

$$I_t(\mathbf{x}) \approx I_{t+1}(\mathbf{x} + \mathbf{w}(\mathbf{x})) \tag{2.1}$$

Prior to resolving this nonlinear equation in $\mathbf{w}$, the Taylor expansion is likely employed for the system linearisation. We have

$$I_t(\mathbf{x}) = I_t(\mathbf{x}) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + H.O.T. \tag{2.2}$$

By considering the first-order components, we have:

$$I_x u + I_y v + I_t = 0 \tag{2.3}$$

where the equation presents the *Optical Flow Constraint* [55]. The subscripts x, y and t present the partial derivatives $\left\{\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t}\right\}$ respectively in spatio-temporal domain. However this linearisation is valid only for small displacements where the image property changes along the motion linearly. In the real-world scenes, the *Brightness Constancy* is often violated by the difficult cases of non-Lambertian reflectance, illumination changing or occlusions. Nevertheless, the constraint often leads to an ill-posed multiple matching problem where a pixel in the first frame matches multiple pixels in the next frame with the similar brightness information. It is mathematically because that two unknowns cannot be determined in a single equation. The extra spatial information from the local or global neighbours is needed to provide a unique solution of the *Optical Flow Constraint*.

2.1.1 Pairwise Optical Flow Models

Local methods, e.g. Lucas and Kanade [87], assume that the optical flow can be simply defined in a local neighbourhood following a parametric form [132, 11, 80]. This approach allows optical flow estimation in each pixel by minimising the convolution of a local Gaussian window G_ρ (with standard deviation ρ) and the constraint.

$$E(\mathbf{w}) = G_\rho \otimes (I_x u + I_y v + I_t)^2 \tag{2.4}$$

Although local methods provide high performance in texture-rich regions [87] and locally simple motion cases, it is often affected by the size of the local window. The smaller window may lead to the local minima issue while the larger window integrates more pixels around but may include the pixels crossing different motion regions [12]. Choosing the right window size is one of the key issues [83, 18, 39, 154, 95] for the local methods and their extensions, which are beyond the scope of our work.

In contrast with local methods, global methods assign a dense vector field for all the pixels and resolve this vector field based on the global information of the image.

Global methods attempt a global smoothness on the local behaviour of the pixel motion – where the neighbouring pixel is assumed to come from the same object surface and behave in a similar way as follows:

$$\begin{aligned} \mathbf{w}_t(x,y) &\approx \mathbf{w}_t(x+1,y) \quad \mathbf{w}_t(x,y) \approx \mathbf{w}_t(x-1,y) \\ \mathbf{w}_t(x,y) &\approx \mathbf{w}_t(x,y+1) \quad \mathbf{w}_t(x,y) \approx \mathbf{w}_t(x,y-1) \end{aligned} \tag{2.5}$$

Horn and Schunck [55] is known as pioneering global approach which introduces a framework by combining a data term (E_{Data}) and a regularisation term (E_{Reg}). Their energy function can be formulated as the weighted sum of two terms:

$$\begin{aligned} E(\mathbf{w}) &= E_{Data}(\mathbf{w}) + \alpha E_{Reg}(\mathbf{w}) \\ &= \int_\Omega \left\{ \underbrace{(I_x u + I_y v + I_t)^2}_{\text{Data Term}} + \alpha \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{Regularisation Term}} \right\} d\mathbf{x} \end{aligned} \tag{2.6}$$

where the higher order regularisation term E_{Reg} encodes a smoothness constraint on the flow vectors $\mathbf{w}$. Although this high order term yields additional difficulties into the minimisation of the energy function, the variational form of the main energy is well defined and tractable to solve. Hence, variances of Horn and Schunck's framework are widely involved in other high performance optical flow estimation algorithms [152, 126, 21] in the last decade. However, the basic *Brightness Constancy* may be violated when the illumination changes temporally through the image sequence. This situation often happens in real-world images with the lighting changing. To deal with this non-constant illumination issue, Brox *et al.* [20] introduce a *Gradient Constancy* assumption where the gradient of the pixel intensity is attempted invariant.

$$\nabla I_t(\mathbf{x}) \approx \nabla I_{t+1}(\mathbf{x} + \mathbf{w}(\mathbf{x})) \tag{2.7}$$

The equivalent linearised form reads:

$$\begin{aligned} I_{xx} u + I_{xy} v + I_{xt} &= 0 \\ I_{xy} u + I_{yy} v + I_{yt} &= 0 \end{aligned} \tag{2.8}$$

Note that similar to Eq. (2.3), the subscripts represent the second order derivatives. A gradient constancy assumption is originally proposed [136] for the aperture issue in the local methods. Such a problem is addressed by the smoothness constraint of the global methods. Brox *et al.* raise it as an invariant into the data term in order

to reduce the dependency on the *Brightness Constancy* assumption when the illumination changes. Combining both the *Brightness Constancy* and *Gradient Constancy* assumptions, the improved data term is formulated as follows:

$$E_{Data}(\mathbf{w}) = \int_{\Omega} \left\{ \underbrace{(I_x u + I_y v + I_t)^2}_{\text{Brightness Constancy}} + \beta \left(\underbrace{(I_{xx} u + I_{xy} v + I_{xt})^2 + (I_{xy} u + I_{yy} v + I_{yt})^2}_{\text{Gradient Constancy}} \right) \right\} d\mathbf{x} \tag{2.9}$$

where the parameter β is a linear weight to control contributions of the *Gradient Constancy* term. This additional assumption is reported to bring extra robustness against the illumination [20, 21]. However, the global energy along with *Gradient Constancy* term is still less robust to motion discontinuities or outliers, e.g. image noisy. In Sec. 2.1.2, we give more details for optical flow estimation on specific objects e.g. water and deformable surface. We also conduct further investigations into existing robust approaches for motion discontinuities and image noisy in Sec. 2.1.3 and 2.1.6.

2.1.2 Geometric Priors

Apart from the brightness based constraints, the geometric priors are considered against specific tracking issues where the motion behaviour of involved objects is physically well studied. Li *et al.* [73, 99] introduce a mass-conservation prior into optical flow in order to trade off the effect of the volume change and depth varying in the textureless water surface tracking. Furthermore, Glocker *et al.* [51] present a prior on local affine motion using the warping behaviour of a triangle mesh. Volz *et al.* [141, 125] propose a temporal coherence prior on dense pixel-trajectories by involving multiple images in the time dimension. Such a prior and the similar 2D subspace constraints [49, 48] are reported efficient in the nonrigid scenario. More details for those methods are given in Sec. 2.1.6. Similar to global brightness based constraints [112, 113, 93, 121, 92], optical flow energy together with the geometric priors often yields the same discontinuity issues in the proposed optical flow fields. Therefore, one possible solution recovering the discontinuity issue is to sufficiently penalise the violation of smoothness constraint.

2.1.3 Regularisation Term

The real-world images may contain a number of object boundaries and outliers. Such difficulties lead to large wrong energy to the main optical flow energy. Horn and Schunck [55] employ the quadratic penalty function (ℓ_2 norm) to penalises deviations. Such quadratic regularisation term corresponds to Gaussian assumption and is often violated in practice, particularly the motion boundaries and occlusions. In such cases, pixels are visible in the current frame but underlying in the next. To overcome

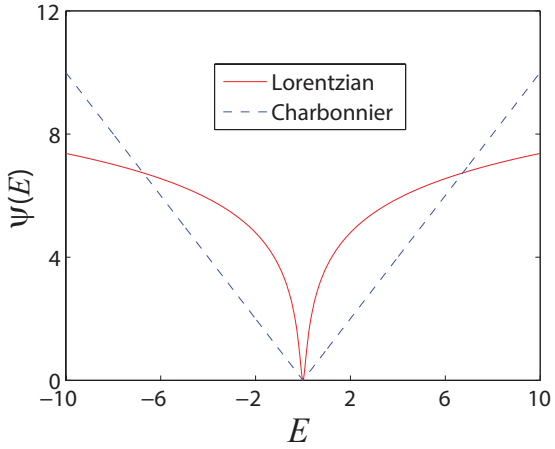


Figure 2-1: Penalty functions Lorentzian and Charbonnier. For better observation, the parameter setting for this plotting refers to Volz *et al.* [141].

these limitations, a great afford has been made on the *Regularisation Term* in order to increase the robustness on the motion boundaries and outliers. Shulman and Herve [120] bring the Huber minimax penalty into the regularisation term against the motion boundaries. Black and Anandan [12] present a framework with an arbitrary penalty function in both data and regularisation term, in particular an equivalent form of the Lorentzian penalty function in their illustrations. Given a penalty function ψ, we have a new energy with penalty on the data term and regularisation term as follows:

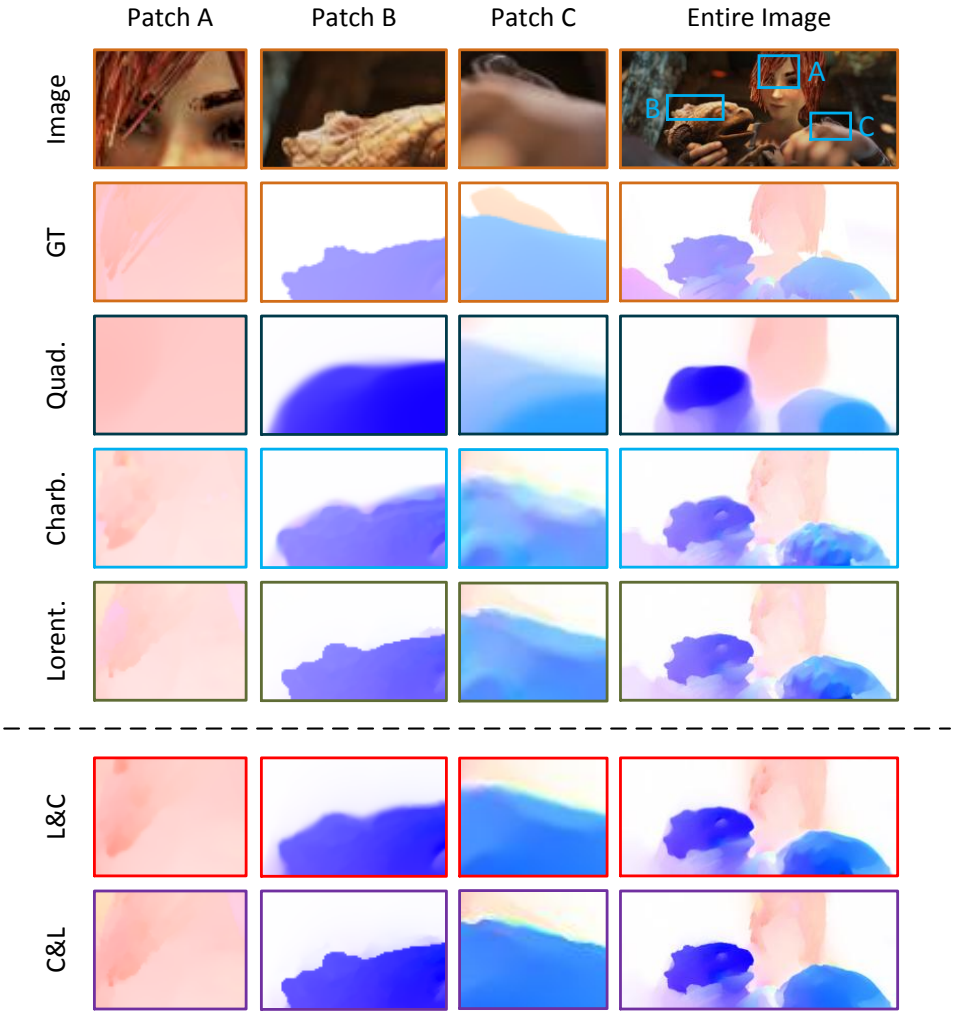
$$\begin{aligned} E(\mathbf{w}) &= \psi_D(E_{Data}(\mathbf{w})) + \alpha\psi_R(E_{Reg}(\mathbf{w})) \\ &= \int_\Omega \psi_D \left(\underbrace{|I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})|^2}_{\text{Brightness Constancy}} + \beta \underbrace{|\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x})|^2}_{\text{Gradient Constancy}} \right) d\mathbf{x} \\ &+ \alpha \int_\Omega \psi_R \left(\underbrace{|\nabla u|^2 + |\nabla v|^2}_{\text{Regularisation Term}} \right) d\mathbf{x} \end{aligned} \tag{2.10}$$

where ψ_D and ψ_R denote penalty functions for data term and regularisation term respectively. They can be either identical or different. The hidden rationale of penalty function is to balance the energy contribution of the terms – e.g. either raises the energy at the outliers or penalises the energy on the boundaries. However such penalty functions – particularly strong non-convex one – may also introduce more difficulties into the energy minimisation. Here we investigate into two well-known penalties and give quantitative analysis on them within the real-world images.

One choice is the ℓ_1 norm, $\psi(x) = |x|$ which is introduced to optical flow regularisation by Aubert *et al.* [4]. ℓ_1 regularisation is widely adopted in the current state-

Average Endpoint Error	Time (Sec.)	Entire Image	Patch A	Patch B	Patch C
Quadratic (Quad.)	23.64	2.21	1.92	2.63	2.85
Charbonnier (Charb.)	25.33	1.86	1.69	2.01	2.33
Lorentzian (Lorent.)	41.12	**1.73**	1.71	**1.80**	**1.86**
Lorent.+Charb. (L&C)	33.93	1.78	**1.67**	1.84	1.90
Charb.+Lorent. (C&L)	36.36	1.84	1.70	1.83	1.88

(a) Quantitative comparison of various implementations on the test frame and patches. The time is recorded for the computation on entire image only.



(b) Visual comparison of various implementations on the test frame and patches.

Figure 2-2: Experimental evaluation of penalties Quadratic, Charbonnier and Lorentzian on sample frame of *bandage_2*, Sintel dataset.

of-the-art approaches [143, 146, 157] because ℓ_1 norm is able to take into account the motion discontinuity. Together with the *Total Variation* optimisation, the approach is also capable to give real-time performance [157]. However, by applying ℓ_1 regularisation, both data term and regularisation term are not continuously differentiable. One possible solution is to replace the ℓ_1 regularisation ($\psi(x) = |x|$) with a differentiable approximation as

$$\psi(s^2) = \sqrt{s^2 + \epsilon^2} \tag{2.11}$$

where ϵ denotes a small constant e.g. 0.001. This convex function is also known as *Charbonnier* penalty that makes the energy easier to solve [126]. Another well-known regularisation is *Lorentzian* penalty with a form as

$$\psi(s^2) = \log(1 + s^2/2\sigma^2) \tag{2.12}$$

where σ is a scale parameter. *Lorentzian* penalty is reported [12, 13] efficient to maintain brightness and spatial agreements while keeping motion discontinuity in the regularisation term and outliers in the data term.

To experimentally evaluate the effect of various regularisation terms, we use quadratic, Charbonnier and Lorentzian on the energy Eq (2.10). In this case, we build five energy functions as: (1) apply quadratic penalty on both data term and regularisation term (**Quad.**); (2) apply Charbonnier penalty on both data term and regularisation term (**Charb.**); (3) apply Lorentzian penalty on both data term and regularisation term (**Lorent.**); (4) apply Lorentzian for the data term and Charbonnier for the regularisation term (**L&C**) and vice versa (**C&L**). Here we set parameters $\epsilon = 0.001$ for Charbonnier; $\sigma = 0.03$ of Lorentzian for regularisation term and $\sigma = 0.1$ for data term; $\alpha = 0.75$ and $\beta = 0.6$ for the main energy function. These parameter settings are fixed throughout the experiment in this subsection. All five energy functions are then minimised using the same scheme mentioned in Sec. 2.1.4. Here the test frame we choose is from *bandage_2* sequence (frame 14) of Sintel benchmark which includes sharp motion boundaries, geometric blur and other advanced features (Sec. 2.1.7). In this experiment, we apply all baselines on the whole frame and calculate the *Average Endpoint Error* (AEE) measure which represents the average Euclidean distance between the endpoints of the baseline optical flow field and the ground truth. More details for the optical flow measures can be found in Sec. 2.1.7. For better observation, we highlight computation time as well as the AEE measures on three patches: (1) **Patch A** contains good texture and smooth motion; **Patch B** involves clear motion boundary; **Patch B** represents blurry boundary.

In Tab. 2-2(a), it is observed that using Lorentzian regularisation for both terms (**Lorent.**) yields best performance on both cases of clear boundary (**Patch B**) and blurry boundary (**Patch C**). The baselines **Charb.** and **L&C** that involved Charbonnier for the regularisation term, result in lower errors than the others in the smooth case (**Patch A**). Furthermore, the mixed regularisation options **L&C** and **C&L** yield competitive performance to each other in all the trials. But **L&C** outperforms the **Charb.** in **Patch A** and faster than **Lorent.** over all. It is because that Lorentzian function provides overall good robustness against outliers (for data term) and boundaries (for

regularisation term) but its non-convexity leads to additional difficulties into the minimisation. Therefore, choosing regularisation is experimental. Lorentzian penalty is preferable for scenes with rich boundaries while Charbonnier penalty is suitable for the smooth case e.g. single object deformation. The mixture – Lorentzian penalty for data term; Charbonnier penalty for regularisation term – intuitively gives good accuracy and less computational consumption.

Apart from the Charbonnier and Lorentzian, *Huber* norm is adopted in some high performance approaches [147, 49]; and often defined as follows:

$$\psi(s^2) = \begin{cases} \frac{s^2}{2\epsilon} & \text{if } s \leqslant \epsilon \\ s - \frac{\epsilon}{2} & \text{otherwise} \end{cases} \tag{2.13}$$

where ϵ is again a small constant. The Huber norm above is considered as a convex differentiable function which performs quadratic regularisation on the small magnitudes ($s \leqslant \epsilon$) but *Total Variation* regularisation otherwise. Intuitively, the Huber norm can provide smoothness constraint on the continuous regions (small magnitudes) and preserve the discontinuity on the motion boundaries (large magnitudes). Please note that the quantitative result of Huber norm is not available in the comparison because this leads to a non-convex energy that cannot be solved by the minimisation scheme mentioned in Sec. 2.1.4.

2.1.4 Energy Minimisation

In this section, we discuss energy minimisation scheme for the variational energy Eq (2.10) where the same penalty function ψ is applied to both data term and regularisation term. In this case, the global energy model is built to meet the Euler-Lagrange equations. For better description, we refer to the common abbreviations [20, 21] of derivatives as follows:

$$I_x = \tfrac{\partial}{\partial x} I_2(\mathbf{x}+\mathbf{w}) \quad I_y = \tfrac{\partial}{\partial y} I_2(\mathbf{x}+\mathbf{w}) \quad I_t = I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x})$$

where the I_1 and I_2 denote the current image and the next image respectively. The further minimiser with Euler-Lagrange equations can be formulated as follows:

$$\begin{aligned} &\psi'(I_t^2 + \beta(\frac{\partial I_t}{\partial x})^2 + \beta(\frac{\partial I_t}{\partial y})^2) \cdot (I_x I_t + \beta(\frac{\partial I_x}{\partial x}\frac{\partial I_t}{\partial x} + \frac{\partial I_x}{\partial y}\frac{\partial I_t}{\partial y})) \\ &\qquad -\alpha\psi'(|\nabla u|^2 + |\nabla v|^2) \cdot \nabla u = 0 \\ &\psi'(I_t^2 + \beta(\frac{\partial I_t}{\partial x})^2 + \beta(\frac{\partial I_t}{\partial y})^2) \cdot (I_y I_t + \beta(\frac{\partial I_y}{\partial x}\frac{\partial I_t}{\partial x} + \frac{\partial I_y}{\partial y}\frac{\partial I_t}{\partial y})) \\ &\qquad -\alpha\psi'(|\nabla u|^2 + |\nabla v|^2) \cdot \nabla u = 0 \end{aligned} \tag{2.14}$$

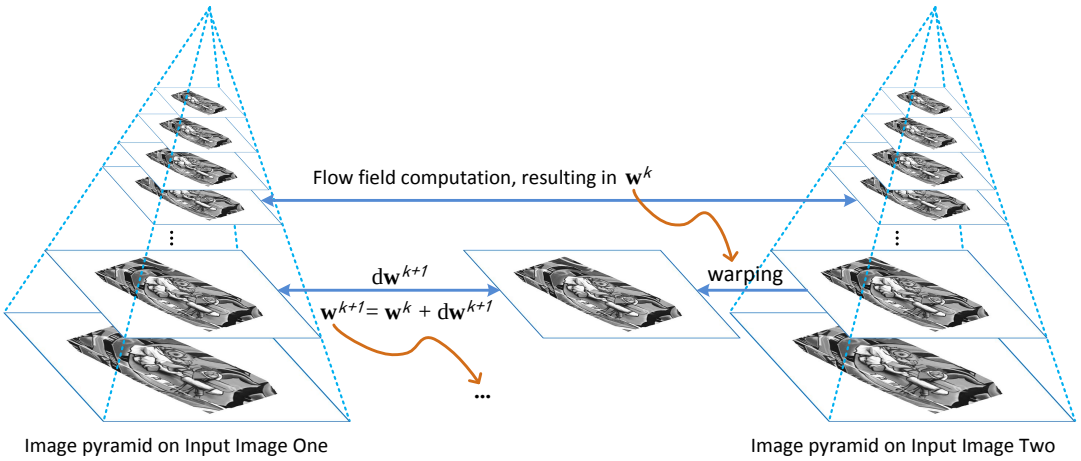


Figure 2-3: Pipeline of warping based coarse-to-fine minimisation framework.

where $\psi'(\cdot)$ presents the derivative of penalty function ψ. This discrete system is still nonlinear in $\mathbf{w}$ because of the argument nonlinearity of the Euler-Lagrange equations, as well as the nonlinearised data term and regularisation term. In this case, it is more difficult to resolve the equivalent linear system of the energy functional Eq. (2.10) by adopting the normal local approaches because of the nonlinearity and the potential local minima. As shown in Fig. 2-3, a common approach in the literature [2, 15, 10] is to employ a warping based coarse-to-fine approach where we first construct Gaussian image pyramids on both input images. On each level, the second image is warped towards the first one based on the flow field propagated from the previous level. The increments between image one and the warped image two is computed before added to the flow field propagated from the previous level. On the coarse level, the linear assumption for local minimisation holds because the more smoother image leads to a lack of the small image details which often results in the local minima. In such case, the first order Taylor expansions can be adopted for linearisation w.r.t Eq. (2.3).

$$I_{*t}^{k+1} \approx I_{*t}^{k} + I_{*x}^{k} du^{k} + I_{*y}^{k} dv^{k}$$

where k denotes the level index of the image pyramid and $* \in \left\{ \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right\}$ denotes the derivatives in the horizontal and vertical directions. However, the small image details are often responsible for the significant information in precise tracking, in particular the high resolution cases. This warping based strategy together with inner level incremental assumption is theoretically justified [19] to take into account the balance between the local minima avoidance and small flow details preservation. It has become popular in recent state-of-the-art methods [152, 80]. Note that minimising the energy Eq. (2.14) in warping based coarse-to-fine framework is well studied. The common numerical schemes for such a energy could be the two nested fixed point iterations proposed by

Brox *et al.* [20]. More derivations can be found in Appendix A.3.

2.1.5 Implementation

Although the risk of local minima is reduced, some implementation tricks are reported important in resolving the energy e.g. scaling parameter in image pyramid [97] and the pre-filtering step for the violation of brightness constancy [143, 97]. Most of such details are responsible for the influence of common difficulties such as the motion boundaries and occlusions which will be discussed later in Sec. 2.1.6.

For a common choice, filtering is widely adopted as a pre-process. Gaussian low-pass filter is applied to reduce the general noise [22, 78] before the optical flow estimation. The *Gradient Constancy* [20] is also considered as a high-order filtering in the minimisation. Median filter [126, 128, 62, 64] is performed on the inner flow field in order to remove the outliers. Furthermore, feature technique (detection and matching) is introduced as a supplement to the optical flow framework. Xu *et al.* [152] adopts the feature correspondences into the flow initialisation to reduce the further risk of local minima.

2.1.6 Common Difficulty and Nonrigid Deformation Challenge

Although robust to energy outliers to a certain extent, such optical flow approaches based on Eq (2.10) are still difficult to recover the correct results from motion boundaries and occlusions. In such regions – as shown in Fig 2-4, lack of the certain information, as well as the dominance of the outliers lead to additional obstacle for a satisfactory solution to the model. Besides, the extra difficulty also arises by large nonrigid deformation. In these cases, adopting robust constraints is a common choice.

Motion Boundary

As mentioned in Sec 2.1.3, the additional penalty provides global regularisation to optical flow estimation on the motion discontinuity. Some authors make a step forward to detect the motion boundaries in specific scenes with multiple objects. In this case, the static edge information may give strong spatial evidences because different objects tend to have different motion behaviour. The motion boundaries are more easily observed on the object boundaries in the image. Taking the image edge property into account, Nagel and Enkelmann [94] particularly smooth the optical flow vectors along the object edge direction but allow motion discontinuities in the orthogonal direction. More recently, Zimmer *et al.* [159] extend this strategy to allow joint image property and optical flow optimisation by introducing the local motion behaviour constraint i.e. local orientations. Wedel *et al.* propose a structure-dependent prior from input images

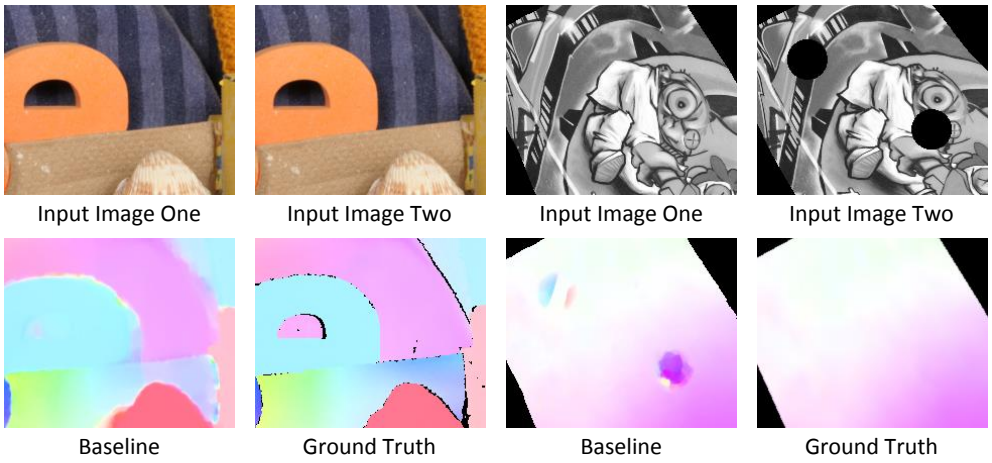


Figure 2-4: Baseline results of optical flow estimation in motion boundaries and occlusions.

into the smoothness in order to preserve motion boundaries. Such an edge/image based strategy is also widely adopted in current state-of-the-art methods [152, 27].

Occlusion

The pixels in real-world images may be occluded or repeatedly appear over time. Such occlusion apparently violates the *Brightness Constancy* and leads to the unpredictable norm errors in the optical flow energy. In the literature, the common solution to detect occlusions [3] in monocular images is to verify the symmetry of optical flow fields in both forward and backward directions where the pixel in current frame performs unique matching to a single pixel in the next frame without any occlusion. Another solution for labelling the occlusions is proposed by Xu *et al.* [152]. Each pixel from current frame is assumed to match at most one pixel in the previous frame otherwise the pixel is occluding or occluded. Such methods can efficiently detect the occlusions but cannot distinguish the occluded pixels from the occluding ones. Some learning-based approaches [124, 58] train the occlusion detector based on appearance cues even the manually labelled landmarks. Sundberg *et al.* [131] consider the gradient changes on both sides of object boundaries to constrain the occlusion detector. To show benefits of occlusion detection in the optical flow estimation, Sun *et al.* [128, 129, 127] combine the optical flow constraint and layered model to obtain top tier performance in the scenes with occlusions.

Although the upon methods show the success in some specific cases, they treat each consecutive pair of images as an independent problem, which weakens their ability in scenes with occlusion. Ricco and Tomasi propose a method that follows traditional cues above: (1) to verify the symmetry (multiple points pixels are mapped to a single one); (2) to verify brightness constancy assumption; but takes into account multiple frames. In their work [103], they express temporal paths of points using a parameterisation

form of low-dimensional path basis. Given a n-frame image sequence $\{I_1, \ldots, I_n\}$, we have:

$$\mathbf{x}_n = \mathbf{x}_1 + \sum_{i=1}^{K} \boldsymbol{q}_i(n) L_i(\mathbf{x}_1) \tag{2.15}$$

where $\mathbf{x}_1 = (x, y)^T$ denotes any point in image I_1; $\mathbf{x}_n$ represents another point in image I_n, which is correspondent to $\mathbf{x}_1$ over time; $\boldsymbol{q}_1(n), \ldots, \boldsymbol{q}_K(n)$ denotes K path basis; $L_i(\mathbf{x})$ states coefficients for the linear combination, which depends on $\mathbf{x}$. They then introduce a constancy that $I_1(\mathbf{x}_1) = I_n(\mathbf{x}_n)$ holds if the point $\mathbf{x}_n$ is visible (not occluded) in image I_n; Here the path basis $\boldsymbol{q}_i(n)$ is supposed to be learned/infered from the sequence. One possible way is to track a sparse point set through the sequence using frame-to-frame tracker e.g. KLT [87, 137]. A set of paths basis (containing K paths) is then estimated using PCA on them. Once the basis $\boldsymbol{q}_i(n)$ is obtained, a energy function is defined to penalise the changes of visible points and the motion difference of nearby points. Their method is supposed to give more accurate prediction on the task of labeling occluded pixels because multiple frames may give accumulating evidence to distinguish occlusions from the violation of brightness constancy.

However it is difficult to obtain high quality paths basis in practice because some tough occlusions or image noise may lead to temporal absences of features. Such issue often results in a sparse set of paths basis for their method. In the extension work [104] of Ricco and Tomasi, they propose an approach to estimate sequence-specific paths basis. They first parameterise features using a $2m \times n$ matrix M – one column per feature – where m denotes the number of frames; n presents the number of features. For instance, point $\mathbf{x}_i = (x, y)^T$ in frame k is entry $(2k-1, i)$ and $(2k, i)$ of M for x and y coordinates respectively. Based on this matrix representation, a feature that reappears after temporary absence may be regarded as a new tracked feature by the tracker, which yields a new column in M. Such issue causes M sparse and further difficult for factorisation. They propose a scheme to factor and compact M by merging groups of columns. They then track a representative point set by typical tracker and maintain the history record of points that have been previously seen but are lost in current frame. Such record is used to align the merged feature paths to each other, resulting in a refined set of paths basis.

To deal with large self-occlusions, Pizarro and Bartoli [100] propose a keypoint based warp estimation algorithm on locally smooth surface. Given a template, an input image and a set of feature point (SURF [7]) matches in between, their method includes three main steps: (1) feature outliers rejection; (2) self-occlusion regions detection; (3) fold-free warp estimation. They first perform a robust outlier rejection method using local-scale smoothness within a triangle mesh. As shown in Fig. 2-5, they consider a pair of matched features ($\mathbf{p}_i$ from template; $\mathbf{q}_i$ from input image). They warp a

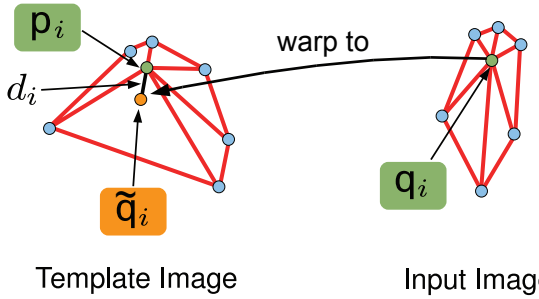


Figure 2-5: Outliers cancellation process of Pizarro and Bartoli [100].

small piece of mesh ($\mathbf{q}_i$ and its neighbours) back to the template; then calculate the distance d_i between the warped point $\tilde{\mathbf{q}}_i$ and the matched template feature $\mathbf{p}_i$. The feature $\mathbf{q}_i$ is considered as an inlier only if the distance d_i is smaller than a threshold. Such threshold can be either predefined or learned from the data. Once high-quality feature matches are obtained, they label regions of the template where such regions may be occluded in the input image. They consider the orientation at each point of the warp. Here the notion of warp orientation can be described by the sign of the warp's Jacobian on the point of the template – negative value means that the point is always occluded. Finally, they give a warp estimation against the self-occlusion. They modify the bending energy by over-smoothing the warp in the self-occluded regions. In this case, such occluded regions are shrunk in the final result. Their pixel-based extension performs high accuracy in both temporal occlusion case and single nonrigid surface case [49]. However, the real-world scenes may involve multiple objects, complex occlusions and image noise. In Chapter 3, we will give more details for how such difficult cases affect the performance of Pizarro and Bartoli [100].

Motion Blur

Motion blur is common photometric effect in real-world images, which often caused by the fast camera movement in a low-light condition. In this case, additional longer exposure time is required in image formation where any slight movement may lead to pixel distortion and colour smear. Strong blur can significantly violate the brightness of the original image, which gives rise to a multiple correspondences issue (Fig. 2-6): a point in the current image corresponds to multiple points in the consecutive image. This uncertainty violates the basic *Brightness Constancy* assumption and gives additional difficulties into optical flow estimation.

In contrary to general cases, rare work is reported in literature to recover optical flow from scenes with blur. The filter flow [116] is proposed to estimate the optical flow along with spatially-varying blur. In such case, those resulting flow field is reported important in image deblurring [54]. Sellent *et al.* [117] capture video sequence using two different frame rates simultaneously then take into account the blur information in the

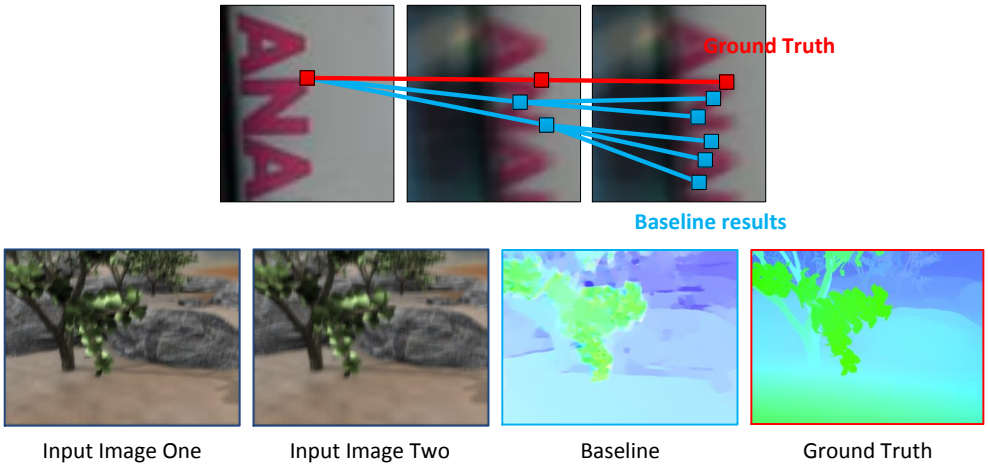


Figure 2-6: Multiple matching issue in blur scene. **Top**: The multiple matchings diffuse on real-world sequence *warrior* (Chapter 4) during the time. **Bottom**: The ground truth sequence is generated by interleaving the Middlebury sequence *Grove2* and a real-world blur kernel (Chapter 4).

longer exposure frames as a constraint to improve the optical flow estimation between shorter exposure frames. Similarly, Liu *et al.* [82] utilise the optical flow of frames between different resolution where a smoothness kernel is obtained in lower resolution frames then applied to higher resolution frames for super resolution video construction. Without the constraints from other image sequence, He *et al.* [53] consider the special feature i.e. corners and obtain the sparse correspondences by hierarchical corner-regions matching where the dense optical flow field is estimated using interpolation. Portz *et al.* [101] propose parameterisation function for both pixel motion and motion-induced blur which is employed to reduce the blur influence during the optical flow estimation. Their method is reported as one of the best optical flow approaches in blurry scenes with similar object depth.

Nonrigid Deformation Challenges

Object deformation is historically classified into rigid or nonrigid categories. As shown in Fig. 2-7, the rigid objects in an image are assumed to be linearly deformed (rotation, translation and scale) to obtain correspondence with respect to the target image. However the correspondence of a nonrigid object between two images cannot be obtained by linear deformation because those object deformation contains nonlinear structure and locally complex motion behaviour in between e.g. local stretching. The presence of such nonrigid deformation within real-world scenes is a challenging issue and leads to two significant difficulties for recovering optical flow. First, the nonrigid deformation often accompanies with large displacement that affects the variational optical flow energy minimisation. Second, the nonrigid deformation is represented as small, locally varying motion that is easily hidden in textureless regions even damaged by the global

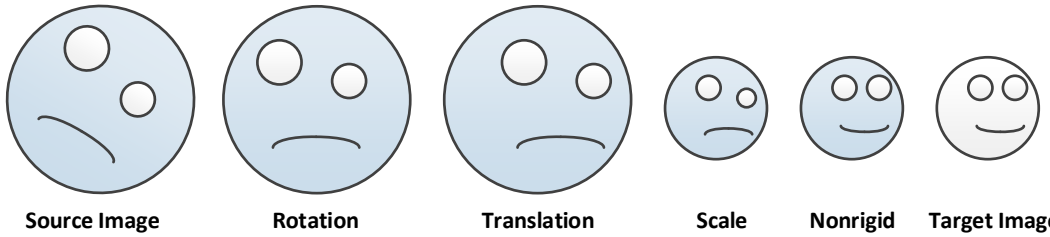


Figure 2-7: Rigid and nonrigid deformation. The source image can be globally rotated, locally translated and scaled but it is different from target image. These differences can be corrected by a series of nonlinear internal deformations.

smoothness constraint.

The large displacement is defined as the unreasonable long distance of the same patch between two images. Such large displacement issue brings unexpected errors to the image derivation (I_x, I_y and I_t) within a variational optical flow model. To deal with this issue, Brox and Malik [21] introduce a descriptor matching term into the variational approach. The main idea is that sparse region-based descriptor matching is supposed to give ability to estimate arbitrarily large displacement. They propose descriptors using histograms of oriented gradients that are sampled on a dense grid in both images. Those descriptors are matched to the nearest neighbours. The mutual best matches are accepted only if the matched patches contain good texture. In this case, each match is assigned a confidence $c(\mathbf{x})$ to describe how good this match is. An indicator function $\delta(\mathbf{x}) = 1$ is also given to locations where successful matches are detected. Such matches are encoded into an additional *Match Term* (E_{match}) as follows:

$$E_{match}(\mathbf{w}, \mathbf{w}') = \int_{\Omega} \delta(\mathbf{x}) c(\mathbf{x}) \psi(\left|\mathbf{w}(\mathbf{x}) - \mathbf{w}'(\mathbf{x})\right|^2) d\mathbf{x} \tag{2.16}$$

where the ψ presents a penalty function while $\mathbf{w}'$ is defined at points where $\delta(\mathbf{x}) = 1$ holds. In this case, $\mathbf{w}'(\mathbf{x}_i) = \mathbf{x}_i - \mathbf{x}_j$ denotes the Euclidean distance of the sparse match $(\mathbf{x}_i, \mathbf{x}_j)$. Once the matches are obtained, their energy is minimised using a coarse-to-fine strategy, a nested fixed point iterations and classic linear system solver (a similar numerical scheme can be seen in Sec. A.3.2). Duo to the good performance for the large displacement case, their method is widely adopted as a baseline in many state-of-the-art work [49, 144, 27]. However, the rigid descriptor HOGs [132] is currently applied in the Brox and Malik's implementation[1]. Such feature descriptor implicitly refers to the local and rigid motion hypothesis, which means that their approach is reliable at salient locations but may deteriorate performance to fast motion and nonrigid deformation. To overcome this drawback, more recent approaches [144, 27] yield competitively higher

[1] http://lmb.informatik.uni-freiburg.de/resources/software.php

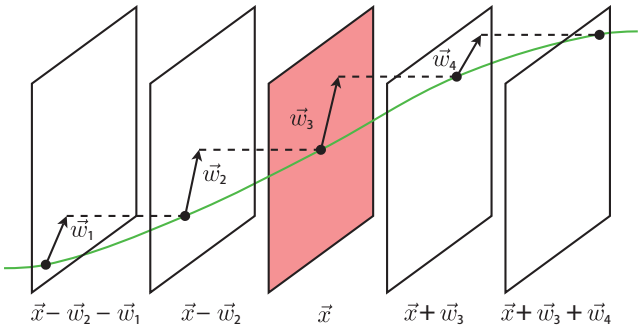


Figure 2-8: Flow parametrisation along motion trajectory over five frames, where $\vec{x}$ presents a pixel location; $\vec{w}_*$ denotes the optical flow vector of $\vec{x}$ in the frame $*$. This image is from Volz *et al.* [141].

performance (in Middlebury and Sintel benchmarks, See the next section) using more sophisticated feature technique e.g. nearest neighbour field and hierarchical patch matching.

Real-world nonrigid objects may perform complex motion i.e. the mixture of large displacement motion and the one that is delicate, small and locally varying. To deal with such difficult issue, Torresani *et al.* [138] introduce a rank constraint together with *Brightness Constancy* to the nonrigid scenario but such method is limited by the local shape changes in large deformation. To describe the motion coherence over time, Volz *et al.* [141] introduce a temporal constraint and a joint spatial constraint into a multi-frame variational model. As shown in Fig. 2-8, they propose a parametrisation of pairwise flow ($\vec{w}_*$) along motion trajectories over five adjacent frames, i.e. all flow fields are registered to a reference frame (the middle frame, red colour one in Fig. 2-8). Based on this parametrisation, they construct complementary regulariser [159] for each pairwise flow field ($\vec{w}_1$, $\vec{w}_2$, $\vec{w}_3$ and $\vec{w}_4$ in Fig. 2-8) separately and combine them into two directional smoothness terms respectively along the orthonormal eigenvectors of the regularisation tensor. The model is then extended by applying a first- and second-order trajectorial regularizations that penalise the change between pairwise flow fields ($\vec{w}_1$, $\vec{w}_2$, $\vec{w}_3$ and $\vec{w}_4$ in Fig. 2-8) over time. Their method shows improvements by considering more frames from long image sequence. However their five-frame approach is less accurate than other top performance pairwise methods (in Middlebury and Sintel benchmarks, See the next section). It is because that their temporal coherence assumption does not always hold particularly in the occluded regions and object boundaries.

In order to solve for the multi-frame registration of deforming surfaces, Garg *et al.* propose a multiple frames based variational approach accompanied with a low-rank subspace constraint. They reuse Ricco and Tomasi's assumption [103] that the temporal movement trajectory of any point can be expressed by a linear combination of a low-rank motion basis. In their early work [47], the unknowns in the variational energy are supposed to be coefficients for motion trajectories over time. Their energy

function combines a data term and a variational smoothness term. The former with a quadratic penalty function penalises the brightness changes along a trajectory; the latter penalises the spatial gradient of the coefficient fields. The energy is minimised using nested fixed point iterations from variational optical flow. In a further extension, Garg *et al.* [49] soften their hard subspace constraints in order to create a prior term as follows:

$$E_{link} = \int_{\Omega} \sum_{n=1}^{F} |\boldsymbol{u}(\mathbf{x}; n) - \sum_{n=1}^{R} \boldsymbol{q}_i(n) L_i(\mathbf{x})|^2 d\mathbf{x} \tag{2.17}$$

where $\boldsymbol{u}(\mathbf{x}; *)$ denotes the trajectory of any point $\mathbf{x} = (x, y)^T$; $\boldsymbol{q}_1(n), \ldots, \boldsymbol{q}_R(n)$ represents R basis trajectories; $L_i(\mathbf{x})$ states coefficients that depend on $\mathbf{x}$ and control the linear combination. In this case, their subspace constraint is softened because $\boldsymbol{u}(\mathbf{x}; n)$ and $L_i(\mathbf{x})$ lead to two similar sets of trajectories but they are not identical. This soft constraint penalises the difference between these two sets in order to yield additional robustness against difficult cases caused by the violation of the brightness constancy assumption. In contrast with other pairwise method, their method aims to obtain an optimal solution across the entire sequence. Thus their energy minimisation takes into account all the frames, then finalises the trajectories of all the points over time. Their method yields high accuracy on the single nonrigid surface where the motion trajectories of points are highly correlated to each other. In Chapter 3 and 5, we will give quantitative evaluations of Garg *et al.* [49] on long nonrigid image sequences.

2.1.7 Benchmarks and Evaluation

Quantitative evaluation on optical flow has been discussed for many years, beginning with the very first benchmark of Barron *et al.* [6] where the synthetic sequences (e.g. the well-known *Yosemite* sequence) are introduced with dense ground truth in between. In their dataset the average angular difference between the baseline optical flow and the ground truth is purposed to represent evaluation metric. These sequences are still widely involved in the current benchmark [5]. As the first well-known benchmark in the community, the sequences are short in length, but the main limitation may lie in a lack of realistic photometric effects, the nonrigid motion and motion blur. This is a fact that such limitations hinder the development of the optical flow community. In the recent years, several new benchmarks are released to the community, which are described in the following sections.

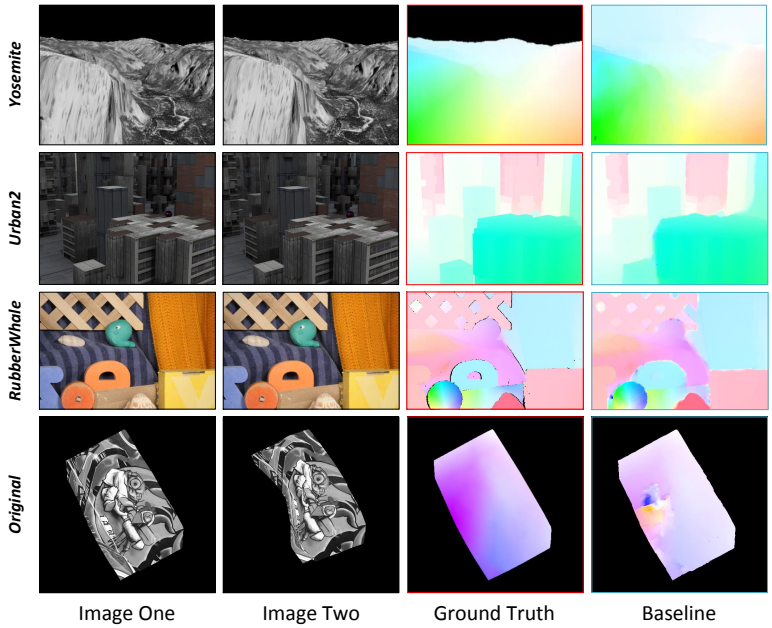


Figure 2-9: Sample frames and ground truth from *Yosemite*, *Urban2*, *RubberWhale* (Middlebury), *Original* (QueenMary). The baseline result is computed by Brox *et al.* [20] for visual comparison.

Benchmarks for Real-world Scenes

Although we may benefit the highly controllable features of the synthetic *Yosemite* benchmarks, the truly real-world sequence is still demanded for more comprehensive evaluation in particular the natural environment. McCane *et al.* [88] bring more realistic images with ground truth. Otte *et al.* [96] capture real-world image with simple objects which make the dense ground truth generation possible even their method is limited in the use of monotonous texture and slight displacement. Bringing the manual notation concept, Liu *et al.* [81] introduce real-world image sequences and ground truth segments where they assume the bad optical flow estimation on the object boundaries but human performs the better distinction for that task. Their benchmarks give dense ground truth for real-world sequences with complex objects but is problematic in the use of evaluation because the major concerns lie in the inconsistency of human performance and the choice of the baseline method for such a solution. Roth *et al.* [105] benefit both real-world and synthetic scenes by generating the ground truth using real-world laser scans and camera motion in rigid scenes.

Consequently the optical flow estimation is heavily limited by the lack of suitable benchmarks until the presence of Middlebury evaluation system. Baker *et al.* [5] proposed a benchmark on real-world sequences associated with dense ground truth. They set a hybrid camera system and a stop-motion scheme to successively capture RGB image and the hidden fluorescent feature map under the ultraviolet. In this case, the ground truth is captured in this hidden feature map then propagated to the RGB im-

Average endpoint error	avg. rank	Army (Hidden texture) GT all	Army (Hidden texture) GT disc	Army (Hidden texture) GT untext	Army (Hidden texture) im0 all	Army (Hidden texture) im0 disc	Army (Hidden texture) im0 untext	Army (Hidden texture) im1 all	Army (Hidden texture) im1 disc	Army (Hidden texture) im1 untext
IVANN [87]	2.6	0.07 1	0.20 2	0.05 1	0.15 1	0.51 3	0.12 5	0.18 1	0.37 1	0.14 1
OFLAF [77]	6.6	0.08 7	0.21 3	0.06 5	0.16 5	0.53 4	0.12 5	0.19 2	0.37 1	0.14 1
MDP-Flow2 [68]	7.8	0.08 7	0.21 3	0.07 14	0.15 1	0.48 1	0.11 1	0.20 4	0.40 4	0.14 1
NN-field [71]	8.4	0.08 7	0.22 14	0.05 1	0.17 7	0.55 6	0.13 10	0.19 2	0.39 3	0.15 6
ComponentFusion [96]	9.8	0.07 1	0.21 3	0.05 1	0.16 5	0.55 6	0.12 5	0.20 4	0.44 7	0.15 6
WLIF-Flow [93]	14.4	0.08 7	0.21 3	0.06 5	0.18 9	0.55 6	0.15 19	0.25 13	0.56 14	0.17 12
TC/T-Flow [76]	14.7	0.07 1	0.21 3	0.05 1	0.19 14	0.68 26	0.12 5	0.28 18	0.66 23	0.14 1
Layers++ [37]	16.1	0.08 7	0.21 3	0.07 14	0.19 14	0.56 9	0.17 26	0.20 4	0.40 4	0.18 18
LME [70]	16.6	0.08 7	0.22 14	0.06 5	0.15 1	0.49 2	0.11 1	0.30 26	0.64 18	0.31 67
nLayers [57]	17.2	0.07 1	0.19 1	0.06 5	0.22 31	0.59 12	0.19 44	0.25 13	0.54 11	0.20 38

Average endpoint error	avg. rank	Mequon (Hidden texture) GT all	Mequon (Hidden texture) GT disc	Mequon (Hidden texture) GT untext	Mequon (Hidden texture) im0 all	Mequon (Hidden texture) im0 disc	Mequon (Hidden texture) im0 untext	Mequon (Hidden texture) im1 all	Mequon (Hidden texture) im1 disc	Mequon (Hidden texture) im1 untext
IVANN [87]	2.6	0.15 1	0.51 3	0.12 5	0.18 1	0.37 1	0.14 1	0.10 2	0.49 3	0.06 2
OFLAF [77]	6.6	0.16 5	0.53 4	0.12 5	0.19 2	0.37 1	0.14 1	0.14 7	0.77 23	0.07 4
MDP-Flow2 [68]	7.8	0.15 1	0.48 1	0.11 1	0.20 4	0.40 4	0.14 1	0.15 18	0.80 29	0.08 10
NN-field [71]	8.4	0.17 7	0.55 6	0.13 10	0.19 2	0.39 3	0.15 6	0.09 1	0.48 2	0.05 1
ComponentFusion [96]	9.8	0.16 5	0.55 6	0.12 5	0.20 4	0.44 7	0.15 6	0.11 3	0.65 6	0.06 2
WLIF-Flow [93]	14.4	0.18 9	0.55 6	0.15 19	0.25 13	0.56 14	0.17 12	0.14 7	0.68 7	0.08 10
TC/T-Flow [76]	14.7	0.19 14	0.68 26	0.12 5	0.28 18	0.66 23	0.14 1	0.14 7	0.86 35	0.07 4
Layers++ [37]	16.1	0.19 14	0.56 9	0.17 26	0.20 4	0.40 4	0.18 18	0.13 6	0.58 4	0.07 4
LME [70]	16.6	0.15 1	0.49 2	0.11 1	0.30 26	0.64 18	0.31 67	0.15 18	0.78 26	0.09 23
nLayers [57]	17.2	0.22 31	0.59 12	0.19 44	0.25 13	0.54 11	0.20 38	0.15 18	0.84 33	0.08 10

Average endpoint error	avg. rank	Schefflera (Hidden texture) GT all	Schefflera (Hidden texture) GT disc	Schefflera (Hidden texture) GT untext	Schefflera (Hidden texture) im0 all	Schefflera (Hidden texture) im0 disc	Schefflera (Hidden texture) im0 untext	Schefflera (Hidden texture) im1 all	Schefflera (Hidden texture) im1 disc	Schefflera (Hidden texture) im1 untext
IVANN [87]	2.6	0.18 1	0.37 1	0.14 1	0.10 2	0.49 3	0.06 2	0.41 1	0.61 1	0.21 2
OFLAF [77]	6.6	0.19 2	0.37 1	0.14 1	0.14 7	0.77 23	0.07 4	0.51 4	0.78 5	0.25 3
MDP-Flow2 [68]	7.8	0.20 4	0.40 4	0.14 1	0.15 18	0.80 29	0.08 10	0.63 14	0.93 14	0.43 15
NN-field [71]	8.4	0.19 2	0.39 3	0.15 6	0.09 1	0.48 2	0.05 1	0.41 1	0.61 1	0.20 1
ComponentFusion [96]	9.8	0.20 4	0.44 7	0.15 6	0.11 3	0.65 6	0.06 2	0.71 27	1.07 32	0.53 29
WLIF-Flow [93]	14.4	0.25 13	0.56 14	0.17 12	0.14 7	0.68 7	0.08 10	0.61 12	0.91 13	0.41 13
TC/T-Flow [76]	14.7	0.28 18	0.66 23	0.14 1	0.14 7	0.86 35	0.07 4	0.67 24	0.98 23	0.49 24
Layers++ [37]	16.1	0.20 4	0.40 4	0.18 18	0.13 6	0.58 4	0.07 4	0.48 3	0.70 3	0.33 6
LME [70]	16.6	0.30 26	0.64 18	0.31 67	0.15 18	0.78 26	0.09 23	0.66 20	0.96 19	0.53 29
nLayers [57]	17.2	0.25 13	0.54 11	0.20 38	0.15 18	0.84 33	0.08 10	0.53 5	0.78 5	0.34 8

Average endpoint error	avg. rank	Wooden (Hidden texture) GT all	Wooden (Hidden texture) GT disc	Wooden (Hidden texture) GT untext	Wooden (Hidden texture) im0 all	Wooden (Hidden texture) im0 disc	Wooden (Hidden texture) im0 untext	Wooden (Hidden texture) im1 all	Wooden (Hidden texture) im1 disc	Wooden (Hidden texture) im1 untext
IVANN [87]	2.6	0.10 2	0.49 3	0.06 2	0.41 1	0.61 1	0.21 2	0.23 2	0.66 2	0.19 1
OFLAF [77]	6.6	0.14 7	0.77 23	0.07 4	0.51 4	0.78 5	0.25 3	0.31 4	0.76 3	0.25 6
MDP-Flow2 [68]	7.8	0.15 18	0.80 29	0.08 10	0.63 14	0.93 14	0.43 15	0.26 3	0.76 3	0.23 5
NN-field [71]	8.4	0.09 1	0.48 2	0.05 1	0.41 1	0.61 1	0.20 1	0.52 45	0.64 1	0.26 9
ComponentFusion [96]	9.8	0.11 3	0.65 6	0.06 2	0.71 27	1.07 32	0.53 29	0.32 6	1.06 19	0.28 12
WLIF-Flow [93]	14.4	0.14 7	0.68 7	0.08 10	0.61 12	0.91 13	0.41 13	0.43 22	0.96 10	0.29 17
TC/T-Flow [76]	14.7	0.14 7	0.86 35	0.07 4	0.67 24	0.98 23	0.49 24	0.22 1	0.82 5	0.19 1
Layers++ [37]	16.1	0.13 6	0.58 4	0.07 4	0.48 3	0.70 3	0.33 6	0.47 33	1.01 13	0.33 34
LME [70]	16.6	0.15 18	0.78 26	0.09 23	0.66 20	0.96 19	0.53 29	0.33 7	1.18 31	0.28 12
nLayers [57]	17.2	0.15 18	0.84 33	0.08 10	0.53 5	0.78 5	0.34 8	0.44 26	0.84 6	0.30 21

Average endpoint error	avg. rank	Urban (Synthetic) GT all	Urban (Synthetic) GT disc	Urban (Synthetic) GT untext	Urban (Synthetic) im0 all	Urban (Synthetic) im0 disc	Urban (Synthetic) im0 untext	Urban (Synthetic) im1 all	Urban (Synthetic) im1 disc	Urban (Synthetic) im1 untext
IVANN [87]	2.6	0.23 2	0.66 2	0.19 1	0.10 4	0.12 6	0.17 11	0.34 1	0.80 4	0.23 2
OFLAF [77]	6.6	0.31 4	0.76 3	0.25 6	0.11 10	0.12 6	0.21 28	0.42 7	0.78 2	0.63 12
MDP-Flow2 [68]	7.8	0.26 3	0.76 3	0.23 5	0.11 10	0.12 6	0.17 11	0.38 3	0.79 3	0.44 4
NN-field [71]	8.4	0.52 45	0.64 1	0.26 9	0.13 29	0.13 25	0.20 22	0.35 2	0.83 5	0.21 1
ComponentFusion [96]	9.8	0.32 6	1.06 19	0.28 12	0.11 10	0.13 25	0.15 7	0.41 6	0.88 9	0.54 5
WLIF-Flow [93]	14.4	0.43 22	0.96 10	0.29 17	0.13 29	0.12 6	0.21 28	0.51 27	1.03 26	0.72 26
TC/T-Flow [76]	14.7	0.22 1	0.82 5	0.19 1	0.11 10	0.11 1	0.30 65	0.50 22	1.02 25	0.64 14
Layers++ [37]	16.1	0.47 33	1.01 13	0.33 34	0.15 50	0.14 45	0.24 40	0.46 12	0.88 9	0.72 25
LME [70]	16.6	0.33 7	1.18 31	0.28 12	0.12 20	0.12 6	0.18 15	0.44 8	0.91 11	0.61 10
nLayers [57]	17.2	0.44 26	0.84 6	0.30 21	0.13 29	0.13 25	0.20 22	0.47 14	0.97 17	0.67 19

Average endpoint error	avg. rank	Yosemite (Synthetic) GT all	Yosemite (Synthetic) GT disc	Yosemite (Synthetic) GT untext	Yosemite (Synthetic) im0 all	Yosemite (Synthetic) im0 disc	Yosemite (Synthetic) im0 untext	Yosemite (Synthetic) im1 all	Yosemite (Synthetic) im1 disc	Yosemite (Synthetic) im1 untext
IVANN [87]	2.6	0.10 4	0.12 6	0.17 11	0.34 1	0.80 4	0.23 2	0.10 4	0.12 6	0.17 11
OFLAF [77]	6.6	0.11 10	0.12 6	0.21 28	0.42 7	0.78 2	0.63 12	0.11 10	0.12 6	0.21 28
MDP-Flow2 [68]	7.8	0.11 10	0.12 6	0.17 11	0.38 3	0.79 3	0.44 4	0.11 10	0.12 6	0.17 11
NN-field [71]	8.4	0.13 29	0.13 25	0.20 22	0.35 2	0.83 5	0.21 1	0.13 29	0.13 25	0.20 22
ComponentFusion [96]	9.8	0.11 10	0.13 25	0.15 7	0.41 6	0.88 9	0.54 5	0.11 10	0.13 25	0.15 7
WLIF-Flow [93]	14.4	0.13 29	0.12 6	0.21 28	0.51 27	1.03 26	0.72 26	0.13 29	0.12 6	0.21 28
TC/T-Flow [76]	14.7	0.11 10	0.11 1	0.30 65	0.50 22	1.02 25	0.64 14	0.11 10	0.11 1	0.30 65
Layers++ [37]	16.1	0.15 50	0.14 45	0.24 40	0.46 12	0.88 9	0.72 25	0.15 50	0.14 45	0.24 40
LME [70]	16.6	0.12 20	0.12 6	0.18 15	0.44 8	0.91 11	0.61 10	0.12 20	0.12 6	0.18 15
nLayers [57]	17.2	0.13 29	0.13 25	0.20 22	0.47 14	0.97 17	0.67 19	0.13 29	0.13 25	0.20 22

Average endpoint error	avg. rank	Teddy (Stereo) GT all	Teddy (Stereo) GT disc	Teddy (Stereo) GT untext	Teddy (Stereo) im0 all	Teddy (Stereo) im0 disc	Teddy (Stereo) im0 untext	Teddy (Stereo) im1 all	Teddy (Stereo) im1 disc	Teddy (Stereo) im1 untext
IVANN [87]	2.6	0.34 1	0.80 4	0.23 2	0.10 4	0.12 6	0.17 11	0.34 1	0.80 4	0.23 2
OFLAF [77]	6.6	0.42 7	0.78 2	0.63 12	0.11 10	0.12 6	0.21 28	0.42 7	0.78 2	0.63 12
MDP-Flow2 [68]	7.8	0.38 3	0.79 3	0.44 4	0.11 10	0.12 6	0.17 11	0.38 3	0.79 3	0.44 4
NN-field [71]	8.4	0.35 2	0.83 5	0.21 1	0.13 29	0.13 25	0.20 22	0.35 2	0.83 5	0.21 1
ComponentFusion [96]	9.8	0.41 6	0.88 9	0.54 5	0.11 10	0.13 25	0.15 7	0.41 6	0.88 9	0.54 5
WLIF-Flow [93]	14.4	0.51 27	1.03 26	0.72 26	0.13 29	0.12 6	0.21 28	0.51 27	1.03 26	0.72 26
TC/T-Flow [76]	14.7	0.50 22	1.02 25	0.64 14	0.11 10	0.11 1	0.30 65	0.50 22	1.02 25	0.64 14
Layers++ [37]	16.1	0.46 12	0.88 9	0.72 25	0.15 50	0.14 45	0.24 40	0.46 12	0.88 9	0.72 25
LME [70]	16.6	0.44 8	0.91 11	0.61 10	0.12 20	0.12 6	0.18 15	0.44 8	0.91 11	0.61 10
nLayers [57]	17.2	0.47 14	0.97 17	0.67 19	0.13 29	0.13 25	0.20 22	0.47 14	0.97 17	0.67 19

Figure 2-10: Screenshot of top ten methods in *Average Endpoint Error* (AEE) test of the current Middlebury ranking (Captured on 22th March 2013).

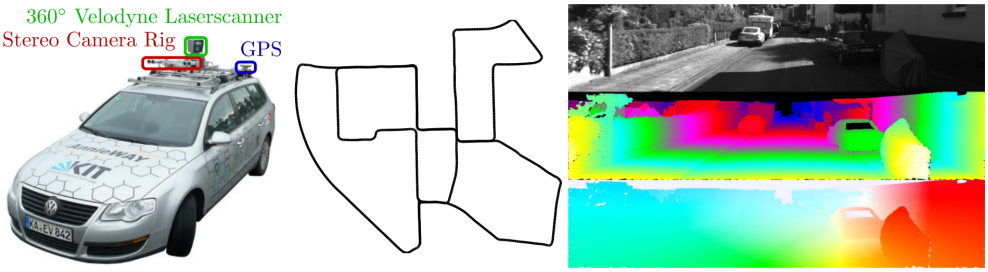


Figure 2-11: *KITTI* dataset: platform setup, sample image, depth and ground truth (the image is from [50]).

age space under the normal lighting. Besides, they provide the ground truth together with the synthetic sequences. Similar to the famous *Yosemite* sequence, their synthetic sequences are generated by computer geometry and texture synthesis technique and contain some difficult features such as large displacement, textureless regions, motion boundaries and large occlusions. To be fair and reduce the risk of over fitting for all the baseline algorithms, they offer the training dataset with ground truth to the potential users while the test dataset contains the RGB images only.

Fig. 2-9 shows the sample frames of several famous benchmarks where the baseline method perform highly accurate on the *Yosemite* sequence but gives large boundary distortion on Middlebury sequences (*Urban2* and *RubberWhale*). Such additional challenges together with the real-world photometric effects result in the rapid development of the optical flow community. However, the top methods on the current Middlebuy list (Fig. 2-10) are tightly ranked where their ranking is strongly affected by a small metric in some specific trials. It is due to the lack of long sequence, challenges of strong illumination change and motion blur, as well as large nonrigid motion.

For more specific driving scenes, the *KITTI* dataset [50, 89] with ground truth is captured using multiviews stereo and calibrated 3D scanner. As shown in Fig 2-11, their platform carries equipments of four high resolution cameras, a Velodyne laser scanner and a localisation system. They drive such a platform around a city to capture consecutive images and 3D scans of the street view. To obtain the high quality pixel correspondence in the image space, they project the accumulated point cloud onto the images; and then match the points across two views to obtain ground truth

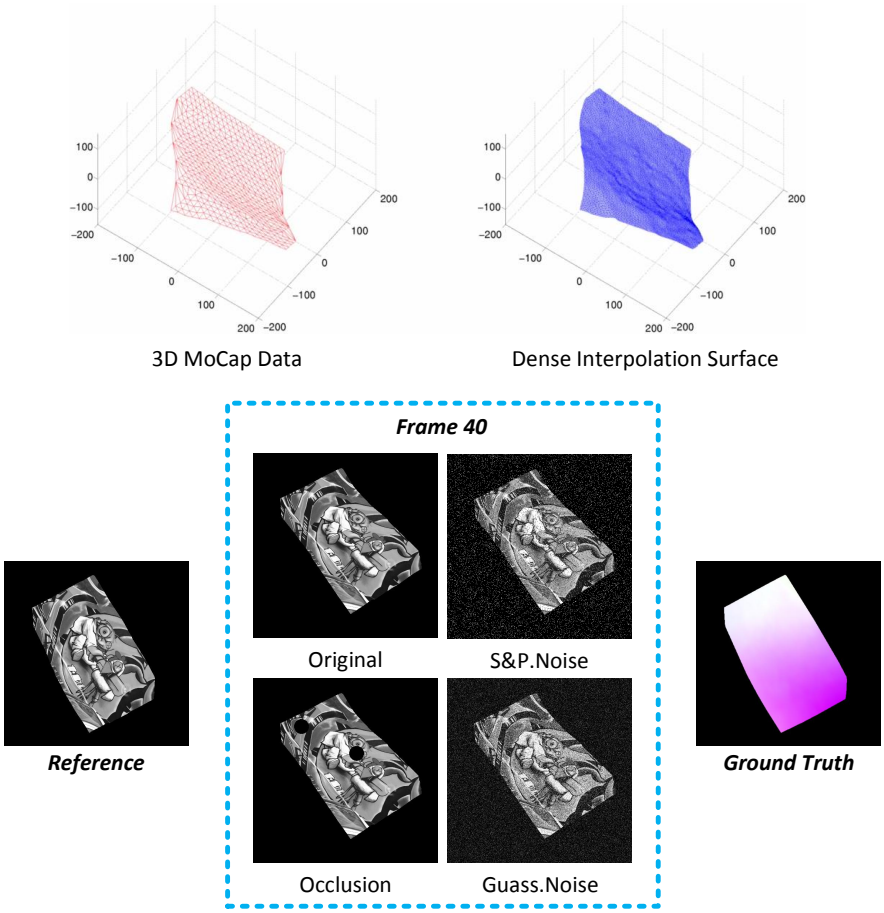


Figure 2-12: Sample frames and ground truth from QueenMary dataset.

correspondences. Here the manual operation is also performed to remove ambiguous image regions e.g. windows and fences. Their dataset contains 194 training and 195 test image pairs. Those images are truly real-world scenes with resolution at 1240 $\times$ 376 pixels. However, Their ground truth is not dense (only 50%). And most of frames from *KITTI* dataset are still, which leads to absence of the motion and scene blur. Comparing to the Middlebury benchmark, *KITTI* dataset comprises real-world sequences for street view but are limited by the low sequence diversity and the lack of nonrigid scenes.

Benchmarks for Nonrigid Motion

Although the Middlebury dataset contains some deformable objects e.g. sequences *Mequon* and *Army*, it is difficult to meet the growing demand of the community. As one of the first attempts for nonrigid specific benchmarks generation, Garg *et al.* [49] propose a long image sequence (QueenMay dataset) with dense ground truth. They capture the sparse 3D *Motion Capture* (MoCap) data of real nonrigidly deformable object then interpolate it for the dense 3D surface. Such a 3D surface is projected

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EpicFlow [2]	4.074	1.332	26.462	3.527	1.030	0.636	0.747	2.087	25.421
TriFlowFused [3]	4.984	1.977	29.508	3.535	1.770	1.529	1.207	2.173	30.568
DeepFlow [4]	5.377	1.771	34.751	4.519	1.534	0.837	0.960	2.730	33.701
IVANN [5]	5.386	1.397	37.896	2.722	1.341	1.004	0.683	2.245	36.342
TriFlow [6]	5.679	2.087	34.980	3.532	1.877	1.821	1.575	2.478	33.845

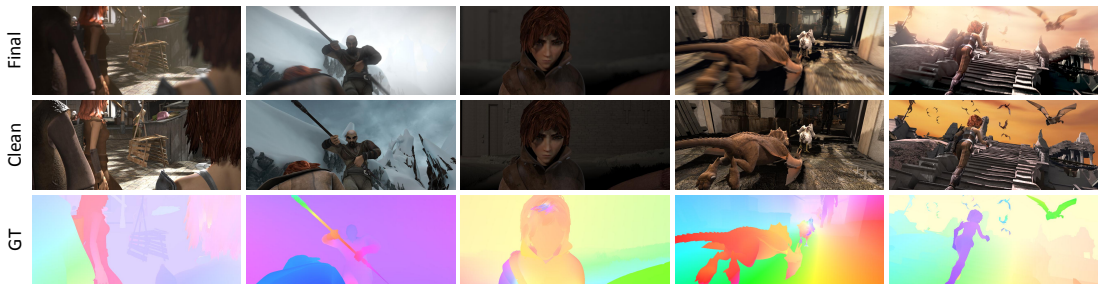


Figure 2-13: Screenshot of top five methods in *Average Endpoint Error* (AEE) test of the current Middlebury ranking (Captured on 22th March 2013). **Top Table**: the ranking of top five approaches in the *Clean* pass. **The Rest**: sample images and ground truth from both *Clean* and *Final* passes.

onto the 2D texture plane in order to synthetically render a long image sequence (60 frames with size of 500×500 pixel) with dense ground truth. To simulate the image noises, the *original* sequence is degraded to three noisy sequences (Fig. 2-12) by adding synthetic occlusions, Gaussian noise and Salt&Pepper noise. The QueenMary dataset contains many exclusive features for nonrigid scenes such as long image sequence, large motion, self-occlusions and dynamic noise. such features lead to its wide adoption in some recent work [32, 48].

Benchmarks for Motion Blur

To overcome the lack of ground truth for motion blur and occlusions, Butler *et al.* [23] simulate *Sintel* benchmarks with the highly naturalistic effects using multiple rendering on a 3D animated short film. They render scenes for different texture conditions by varying complexity. In this case, many comprehensive features are presented in their sequence such as long sequences, specular reflections, very large displacement, motion blur and atmospheric effects.

Fig. 2-13 shows sample frames and the ground truth of *Sintel* dataset. Their dataset contains two categories – *Clean* and *Final* – each of which includes 12 long synthetic sequences. The *Clean* pass mainly contains the various properties of changing illumination, shadow and specular reflections. Such properties give more realism into the synthetic scenes. The *Final* pass contains all sequences from the *Clean* pass but added more difficult atmospheric effects, depth of field blur and motion blur, etc. In general,

Sintel dataset is more difficult than Middlebury and Garg *et al.* (QueenMay dataset) because it represents very large displacement (larger than 40 pixel) and geometric blur. Both of these issues are still unsolved in the optical flow community.

Fig. 2-13 (Top table) shows the evaluation in Sintel dataset where several metrics[2] – EPE, matched, unmatched, d measure (d0-10, d10-60 and d60-140) and s measure (s0-10, s10-40 and s40+) – are performed. EPE denotes the overall *Endpoint Error* which is also widely used in other benchmarks; matched is the *Endpoint Error* on the unoccluded region while unmatched presents the *Endpoint Error* on the occluded one. Apart from these traditional measures, they provide statistic metrics on the occlusion boundaries (d measure), as well as for different displace ranges (s measure).

Sintel dataset is increasing popular in the community. It is observed that some recent baselines that rank high in Middlebury benchmark show relatively low performance in the Sintel dataset. That may give more room for potential new methods.

Evaluation Measures

To measure the accuracy of the optical flow estimation, there are two common ways w.r.t *Endpoint Error* (EE)[5, 96, 49] and *Angle Error* (AE) [6, 42] in the literature. The EE is defined as the Euclidean distance between the baseline optical flow vector $\mathbf{w}$ and the ground truth motion vector $\mathbf{w}_{GT}$ as follows:

$$EE = |\mathbf{w} - \mathbf{w}_{GT}| \tag{2.18}$$

The AE denotes the angel difference between the $\mathbf{w}$ and the $\mathbf{w}_{GT}$. We have

$$AE = cos^{-1}\left(\frac{\mathbf{w} \times \mathbf{w}_{GT}}{|\mathbf{w}| \cdot |\mathbf{w}_{GT}|}\right) \tag{2.19}$$

For the robust test, additional statistics analysis e.g. average and standard deviations, is also performed on the error metrics [5, 23]. Another common measure is known as the interpolation evaluation [5] for the sequences which are captured without the dense ground truth correspondence. In this case, the high-speed camera is operated to capture the continuous frames. Every other frame is provided for evaluation and the intermediate frame is retained as the interpolation ground truth in the image space. Some other metrics e.g. d measure and s measure, are also considered by Sintel dataset. More details can be found in Sec. 3.4.5.

2.2 Nonrigid Surface and Laplacian Operator

In real-world scenes, nonrigid deformation is often observed as highly flexible motion on a soft surface. Such a deformable surface is difficult to represent due to large number

[2] http://sintel.is.tue.mpg.de/results/

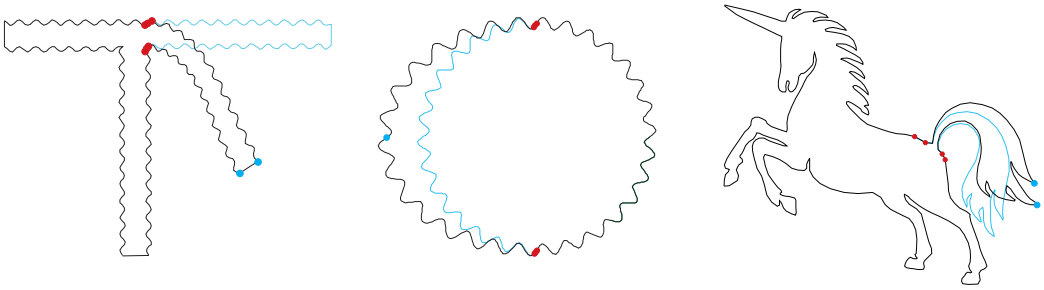


Figure 2-14: Sample mesh deformation using Laplacian mesh processing framework [123]. Fixed control points: **Red Points** are anchor vertices; **Blue Points** are pulled-handle vertices.

of parameters (more than 200 degrees of freedom), which leads to the intricate even unsolvable prior to tracking energy. In the last decade, many work from the graphics community is proposed to represent and edit such a nonrigid surface using triangular mesh geometry.

The early surface representation is parametric based approach [38, 57] using subdivision techniques [115]. However such approaches may limit the type of deformation by varying the value of the parameters. More sophisticated piecewise-linear surface representation is introduced as a triangular mesh that yields intuitive surface display and allows for the use of triangular topological properties on surface deformation analysis. In such mesh representation, the nonrigid deformation is presented as the mesh deformation operation which should naturally modify the shape and simultaneously respect the geometric detail. Welch and Witkin [145] preserve the geometric smoothness during the mesh deformation. The mesh often contains distinguishing features which is also supposed to preserve for realistic deformation simulation [67]. More common approach for mesh deformation is the Laplacian processing framework [122, 123] which is based on linear Laplacian operators defined on triangular meshes.

2.2.1 Laplacian Representation and Processing

Let a triangular mesh $\mathcal{M}$ is presented by $\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ denotes set of the geometric positions of the vertices in absolute cartesian coordinates, $\mathbf{E}$ is edge set, and $\mathbf{F}$ presents the face set. In a connected mesh, the surrounding adjacent vertices is considered as the *neighbourhood ring* of a specific vertex v_i, denoted by $\mathcal{N}_i = \{j \mid (i,j) \in \mathbf{E}\}$ and the number of elements in $\mathcal{N}_i$ is presented by d_i. In this case, the geometric mesh can be presented in an equivalent differentials form as $\Delta = \{\mathcal{L}(v_1), \mathcal{L}(v_2), \ldots, \mathcal{L}(v_n)\}$ which denotes the difference between v_i and the average of its neighbours v_j within the *neighbourhood ring*.

$$\mathcal{L}(v_i) = v_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} v_j \tag{2.20}$$

In the literature, other sophisticated differentials e.g. Cotangent weights [35], can also be adopted. Note that transformation from $\mathbf{V}$ in absolute cartesian coordinates to the differentials Δ can be performed as:

$$\Delta = \mathbf{L}\mathbf{V}, \ \textit{where}\ \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \tag{2.21}$$

where the $\mathbf{L}$ denotes the Laplacian matrix and $\mathbf{D} = \texttt{diag}(d_1, d_2, \ldots, d_n)$ is the degree matrix. $\mathbf{I}$ is considered as an identity matrix and $\mathbf{A}$ denotes the adjacency matrix for the deformable mesh. In this case, the mesh deformation using Laplacian coordinates can be mathematically described as fixing the control points (Red and Blue points in Fig. 2-14) i.e. $\{\mathbf{V}'_c | v'_{c\cdot i} \rightarrow v_i, m \leqslant i \leqslant n, m < n\}$ and locate the remaining vertices $\{\mathbf{V}' | v'_i, 1 \leqslant i \leqslant m-1\}$ by minimising the Laplacian energy:

$$\mathbf{V}' = \underset{v'}{\textbf{argmin}} \left\{ \sum_{i=1}^{n} \|\mathbf{L}v'_i - \mathcal{L}(v_i)\|^2 + \sum_{j=m}^{n} \omega \|v'_{c\cdot j} - v_j\|^2 \right\} \tag{2.22}$$

where the weight ω balances the contribution of the positional constraints of the control points. Given the predefined constraints (a set of control points), this Laplacian based approach is reported efficient to preserve the local shape and small surface details during mesh deformation. However such a Laplacian representation is rarely adopted in variational optical flow model for nonrigid scenario. It is because that Laplacian representation often results in discrete energy that is difficult to minimise within a variational framework. Unsuitable meshes for a scene with multiple nonrigid objects may lead to wrong energies to main energy function. This issue will be further discussed in Chapter 3.

2.3 Image Deblurring

Many challenges in tracking stem from noises severely accompanied real-world images capture. Scene blur, as one of common photometric effects, is the vague phenomenon in the images captured when the sensor is shifted during the exposure. In this case, the blur removal is taken as the important pre-process in the tracking task. The blur information extraction and removal has been discussed for many years [66, 102]. However, a comprehensive literature review on image deblurring is out of scope. In this work, we consider single image deblurring under the spatially invariable blur assumption where the blur is invariable for every pixel of the input image.

2.3.1 Blind Deconvolution

Spatially invariant blur assumption is presented that the blur kernel of an observed image is uniform in spatial domain. In this case, objects within the same scene are

assumed to have similar depth refer to the camera as well as the slow movement. The recovering process can be simplified to allow the blind deconvolution [66] of an unobserved latent image ℓ and a single, spatially invariant blur kernel k from the input blurry image I:

$$I = k \otimes \ell + n \tag{2.23}$$

where the $\otimes$ in between is the convolution operation and $n \sim \mathcal{N}(0, \sigma^2)$ denotes the noise which is commonly assumed as Gaussian noise. The well-known solution for this deconvolution is the Maximum-a-Posteriori based estimation ($\text{MAP}_{\{\ell,k\}}$) [40] which is to seek a pair of $\{\ell, k\}$ to maximise:

$$p(\ell, k|I) \propto p(I|\ell, k)p(\ell)p(k) \tag{2.24}$$

where $p(I|\ell, k)$ denotes a likelihood which is represented as the data fitting term [28]. Both $p(\ell)$ and $p(k)$ favour the natural blurry image as regularisers. The common optimisation process can be formulated as follows:

$$\{\ell, k\} = \textbf{argmin}_{\{\ell,k\}} \left(\underbrace{\lambda \|I - k \otimes \ell\|}_{\text{Data Fitting Term}} + \underbrace{\rho(\ell, k)}_{\text{regulariser}} \right) \tag{2.25}$$

where the energy Eq. (2.25) contains both the data fitting term and potential regularisers. However, this $\text{MAP}_{\{\ell,k\}}$ solution is reported difficult to recover both ℓ and k on small image structure without strong assumptions and suitable priors [28, 29, 61, 151, 59, 119]. Besides the blur kernel k which is often assumed to be sparse [40, 118] and continuous [25], it is assumed that the gradients of the latent image ℓ should be informative and heavy-tailed [71, 72, 65]. Such assumptions fundamentally lead to the fact that the application scope of the algorithm is strongly limited. Furthermore, the intensity boundary (edge) is also considered as important information in blind deconvolution. Cho *et al.* [30] and Joshi *et al.* [61] extract and utilise the sharp edge information from the blur image into the blind deconvolution. Those methods are often reported difficult to handle the large blur which strongly reduce the accuracy of edge restoration [130]. In this case, a further improvement on $\text{MAP}_{\{\ell,k\}}$ based method is proposed to estimate the blur kernel and latent image iteratively, which is known as two-phase iterative framework.

2.3.2 Two-Phase Iterative Deconvolution

The main idea of the two-phase iterative framework is to recover the blur kernel k using the Maximum-a-Posteriori based estimation on kernel (MAP_k). Given this result kernel, the latent image ℓ is obtained by the non-blind deconvolution. This process is

Figure 2-15: Sample results of Xu *et al.* [151], Zhong *et al.* [158] and Cho *et al.* [28] on the blurry image *summerHouse*. Note that 40×40 kernel is employed in Cho *et al.*.

often applied into an iteratively coarse-to-fine fashion [28] where the kernel is computed from an predicted latent image and given blurry image. Such kernel is used to estimate the potential latent image which is propagated to the next iteration for the kernel estimation. Based on Shan *et al.* [118] and the blur kernel sparse prior, Cho and Lee propose a fast deconvolution scheme in two-phase iterative fashion. In their method the energy Eq. (2.25) is explored into two parts for latent image estimation and blur kernel recovery respectively.

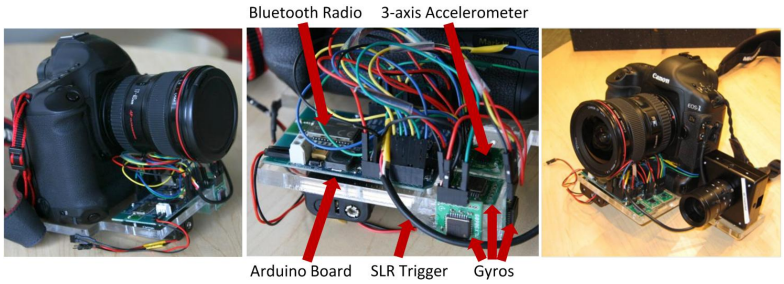
$$E(\ell) = \sum_{\partial_*} \omega_* \left\| k \otimes \partial_* \ell - \partial_* I \right\|_2^2 + \lambda_1 \left\| \nabla \ell \right\|_2^2 \tag{2.26}$$

$$E(k) = \sum_{\partial_*} \omega_* \left\| k \otimes \partial_* \ell - \partial_* I \right\|_2^2 + \lambda_2 \left\| k \right\|_2^2 \tag{2.27}$$

where $\partial_* \in \{\partial_0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ and ω_* denotes the predefined weights for each partial derivative. Both λ_1 and λ_2 present weights for the regularisers. Note that their method use only the image derivatives where the ∂_* is adopted as up to only second order derivatives, which leads to fast convergence in the implementation. In the case, the blur edge can be accurately extracted by padding the boundaries of the derivative map. To deal with the large blur issue mentioned in general blind deconvolution, Cho and Lee [28] bring this improved edge information but interleave the sharp edges restoration and blur kernel estimation on each level of the coarse-to-fine framework. Their method is widely adopted and extended to many modifications [151, 158] as shown in Fig. 2-15 which are reported effective in recent work [63].

2.3.3 Hardware-Aided Approaches

The semantic prior knowledge on the blurry image provide more evidences and make the image deblurring tractable but still limited by the local image statistics. Those approaches often perform with heavy computation consumption and easily fail in cases of large blur or spatially invariant blur. The mixture of camera defocus and shake blur



(a) The camera setup of Levin *et al.* by affixing inertial measurement sensors to RGB cameras.

(b) Visual comparison of Levin *et al.* against Cho *et al.*, Krishnan *et al.* and Xu *et al.*

Figure 2-16: Camera setup of Levin *et al.* [70] and the visual comparison to other image-based approaches i.e. Cho *et al.* [28], Krishnan *et al.* [65] and Xu *et al.* [151].

is also hard to deal with in the recent image-based methods [133, 60]. In such case, the defocus blur is removed along with the unintentional camera-shake blur, which yields over-deblurred image.

In order to address these limitations in image deblurring, hybrid approaches combined hardware sensors and the software, are proposed to take into account both the image space as well as the additional information channel [70]. Ben-Ezra *et al.* [9] and Tai *et al.* [133] propose that the camera motion can be a constraint in image

deblurring. They attach a video camera to a high speed camera in order to capture the same scene simultaneously. The dense inter-frames motion is then computed to estimate the camera motion. Their solution provides high speed deblurring on single image but poor portability in practice. Park *et al.* [98] apply a three-axis accelerator to a camera for motion measurement. Similar to their work, Levin *et al.* [70] develop inertial measurement sensors matrix (Fig. 2-16(a)) attached to a normal camera. Such sensors are lightweight and allow to capture more degrees of camera motion. Such a solution is supposed to be automatic, handling complex blur and more efficient against the image-based approaches (Fig. 2-16(b)).

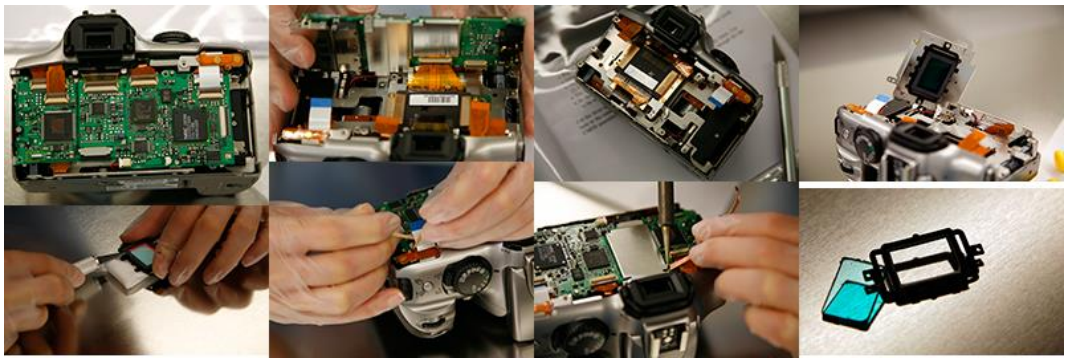
2.4 Near-Infrared Imaging

Visible spectrum is known as the portion of electromagnetic spectrum which can be perceived by the biological vision system of human which can respond to light with wavelengths in the range of approximately 390 to 700 nm. The *Near-Infrared* (NIR) spectrum comprises wavelengths in the range of about 700 to 1100. Although NIR spectrum is just located after the long band of the visible spectrum, it has been reported [149, 44, 86, 107, 17] that varying intensity in visible bands gives rare information about the NIR response (Fig. 2-18). Such a distinguished feature expands the use of NIR imaging into wide application areas such as face-based security [74], remote sensing [79, 140, 85] and scene recognition/classification [91, 17].

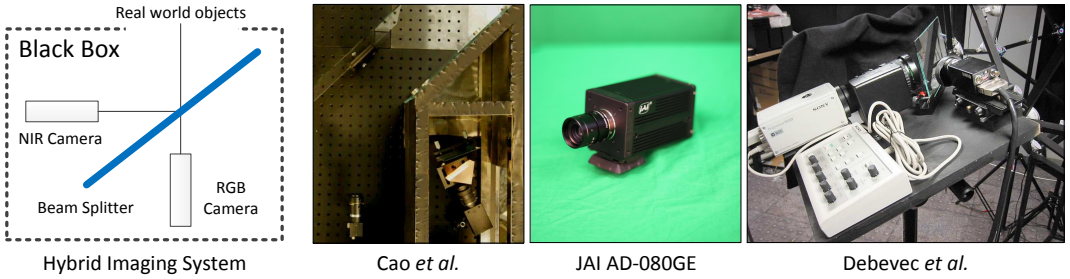
2.4.1 Near-Infrared Image Capture

To capture the NIR spectrum, the NIR films are produced and a common option in the last two decades but their applications are often limited by the strict precautions and the long exposure. In fact, the digital camera with silicon-based sensors (CCD or CMOS) are sensitive over a larger range of approximately 350 to 1100 nm which cover main regions of both visible and NIR spectrum. In this case, a visible spectrum pass filter is applied in front of the sensors in order to prevent NIR signal. Fredembach *et al.* [44] modify a Canon 300D SLR by replacing such a filter by a near clear glass (Fig. 2-17(a)). Their camera can subsequently switch to capture either RGB or NIR images by applying a RGB-pass or NIR-pass filter onto the lens. Their modification allows video recording but is not capable to capture both RGB and NIR images simultaneously from the same scene.

Debevec *et al.* [33] propose a hybrid camera system by interleaving a beam splitter, the RGB camera and NIR camera (Fig. 2-17(b)). In their system, the natural light is equally split into both cameras by the beam splitter at a 45 degree angle in the middle. After the calibration Such a system allows the user to record video sequences in RGB and NIR spaces simultaneously. Replacing the NIR camera by a grayscale camera,



(a) Building an NIR camera by modifying a normal camera [44].



(b) Hybrid camera system to capture both RGB and NIR images from the same scene. **From Left To Right**: The sample systems of Cao *et al.* [24], JAI AD-080GE and Debevec *et al.* [33].

(c) Sample NIR images and their related RGB channels [17].

Figure 2-17: Near-infrared imaging systems.

Cao *et al.* [24] propose a similar imaging system to capture multispectral sequences for object classification and surface tracking. However those beam-splitter-based systems is always difficult calibration and has poor portability. Some commercial products e.g. JAI AD-080GE, pack components into a small camera body but yield low resolution images.

2.4.2 Visible and Near Infrared Spectrums Absorption

The NIR spectrum gives extra information from the visible spectrum (R,G and B bands) because the natural object surfaces have significantly different reflectance for

Figure 2-18: Spectrum reflectance [43] by varying material surfaces.

different spectrum. Fig. 2-18 shows the reflectance of some typical objects in the natural environment. It is observed that most natural objects except the water absorb less spectrum located after 700 nm. In this case those objects are represented more bright in the NIR image which is visibly different from the RGB image. Such an observation drives many NIR relevant work in the computer vision community. Fredembach *et al.* [44] enhance the image colour in visible RGB channel by introduce the NIR channel as either additional colour or luminance components. Schaul *et al.* [111] expose that the long wavelength of NIR give rise to scattering reduction against the haze scene. Besides, Fredembach and Süsstrunk [45] propose a darkness map together with color-to-NIR ratios in order to label the shadow edge from the bright background. Brown *et al.* [17] propose a multispectral SIFT feature by bringing the NIR intensity as the fourth dimension together with the R, G and B bands. Such an improved feature descriptor is reported efficient in scene recognition. Furthermore, the NIR spectrum is important assist to tracking. Yang *et al.* [155] highlight predefined markers on the surgical tool using NIR camera. Such tracked markers are supposed to be significant information for tracking the 3D pose and position of the surgical tool in real time.

2.5 Challenges and Contributions

Based on our reviews, we identify an open research issue of dense nonrigid surface tracking on long real-world sequence. The rest of this thesis illustrates our major contributions to this research challenge as follows:

- In Chapter 3 we first investigate performance of local *Laplacian Mesh Constraint* in order to improve the optical flow estimation on nonrigid surfaces.
- In Chapter 4 we discover novel blur representation between video frames using camera motion trajectory which is then introduced as an additional constraint in order to improve the optical flow robustness in frames of blurry video footage.
- In Chapter 5 we research the *Drift* issue in the long sequence. We also give an optimisation framework against this problem by interleaving the above dense tracking strategy and the long term feature technique.

- In Chapter 6 a multispectral imaging system together with infrared visible dyes is developed to capture the dense ground truth of nonrigid surface for quantitative evaluation and other difficult tracking tasks.

In the following chapters, our major contributions are described in details.

Chapter 3

Pairwise Nonrigid Tracking using Laplacian Mesh Constraint

In this chapter we present a hybrid optical flow algorithm for nonrigid surface tracking. We introduce a novel *Laplacian Mesh Energy* formula to encourage local smoothness whilst simultaneously preserving nonrigid deformation. This unique *Laplacian Mesh Energy* term is expressed wholly within a classic variational optical flow model, and show its efficient optimisation in an improved coarse-to-fine pyramidal approach. We evaluate our approach on the widely recognized *Middlebury* dataset [5] as well as the publicly available nonrigid data set proposed by Garg *et al.* [49]. Our approach provides excellent performance ranked in the top tier of the *Middlebury* evaluation[1], and either outperforms or shows comparable accuracy against the leading publicly available nonrigid approaches when evaluated on the nonrigid data set of Garg *et al.*

3.1 Introduction

Optical flow estimation is an important area of computer vision research. Current algorithms can broadly be classified into two categories – variational methods and discrete optimisation methods. The former is a continuous approach [20, 22, 159] to estimate optical flow based on modifications of Horn and Schunck's framework proposed in [55]. Such approaches can provide high subpixel accuracy but may be limited by minimisation of the non-convex energy function. The latter [15, 139] is based on combinatorial optimisation algorithms such as min-cut and max-flow, which can recover non-convex energy functions and multiple local minima but may suffer from discretisation artifacts, e.g. the optical flow field boundary is aligned with the coordinate axes. One desirable property of optical flow techniques is to preserve local image detail and also handle

[1] http://vision.middlebury.edu/flow/eval/results/results-e1.php

nonrigid image deformations. Under such deformations, the preservation of local detail is particularly important. Garg *et al.* [49] impose this by maintaining correlations between 2D trajectories of different points on a nonrigid surface using a variational framework. Pizarro *et al.* [100] propose a feature matching approach based on local surface smoothness, and also show particular application to nonrigidly deforming objects.

In computer graphics research, a common requirement is that surface meshes are globally editable, but capable of maintaining local details under mesh deformations. In order to provide a flexible representation to allow computation and preservation of such details, Laplacian mesh structures have previously been described [122, 90]. Such schemes impose constraints in differential Laplacian coordinates calculated upon groups of triangles associated with each vertex. Meshes have previously been used in optical flow estimation [51]. However, this is to reduce processing complexity as opposed to specifically imposing smoothness.

In this chapter we present an variational optical flow model which introduces a novel discrete energy based on *Laplacian Mesh Deformation*. Such deformation approaches are widely adopted in graphics research, particularly for preserving local details [122, 90]. In our work we propose that the same concept, i.e. that of an underlying mesh which penalizes local movements and preserves smooth global ones, can be of great use for optical flow and tracking. Constraints on the local deformations – that is expressed in Laplacian coordinates – encourage local regularisation of the mesh. Our algorithm applies a mesh to an image with a resolution up to one vertex per pixel. The *Laplacian Mesh Energy* is described as an additional term for the energy function, and can be applied in a straightforward manner using our proposed minimisation strategy. In addition, a novel coarse-to-fine approach is described for overcoming the loss of small optical flow details during its propagation between adjacent pyramid levels.

3.2 Hybrid Energy

In this section, we introduce our novel hybrid energy formula in which our algorithm considers a pair of consecutive frames in an image sequence. The current frame is denoted by $I_1(\mathbf{x})$ and its successor by $I_2(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$ is a pixel location in the image domain Ω. We define the optical flow displacement between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ as $\mathbf{w} = (\mathbf{u}, \mathbf{v})^T$. In the proposed optical flow estimation approach, the core energy function can be expressed by the following:

$$E(\mathbf{w}) = E_{Data}(\mathbf{w}) + \lambda E_{Lap}(\mathbf{w}) + \xi E_{Smooth}(\mathbf{w}) \tag{3.1}$$

where $E_{Data}(\mathbf{w})$ denotes a data term that expresses both *Brightness Constancy*

and *Gradient Constancy* assumptions on pixel values between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$. Similar to [20, 55], a smoothness term is introduced into the formula, which controls global flow smoothness. The term E_{Lap} represents our core contribution, i.e. the *Laplacian Mesh Energy* $E_{Lap}(\mathbf{w})$. All the three terms are detailed in the following sections.

3.2.1 Continuous Brightness Energy

Following the standard optical flow assumption regarding *Brightness Constancy*, we assume that the gray value of a pixel is not varied by its displacement through the entire image sequence. In addition, we also make a *Gradient Constancy* assumption which is engaged to provide additional stability in case the first assumption (*Brightness Constancy*) is violated by changes in illumination. The data term of energy function encoding these assumptions is therefore formulated as:

$$E_{Data}(\mathbf{w}) = \int_{\Omega} \psi(\|I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x})\|^2 + \theta \|\nabla I_2(\mathbf{x}+\mathbf{w}) - \nabla I_1(\mathbf{x})\|^2) d\mathbf{x} \tag{3.2}$$

For robustness regarding occlusions and boundaries, we apply the Lorentzian as the penalty function $\psi(s) = log(1+s^2/2\epsilon^2)$ to solve this formula. The term $\nabla = (\partial_{xx}, \partial_{yy})^T$ is the spatial gradient and $\theta \in [0, 1]$ denotes a weight that can be manually assigned with different values. Furthermore, the smoothness term of our algorithm is a dense pixel based regularizer that penalizes global variation. The objective is to produce a globally smooth optical flow field:

$$E_{Smooth}(\mathbf{w}) = \int_{\Omega} \varphi(\|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{v}\|^2) d\mathbf{x} \tag{3.3}$$

where we employ Charbonnier penalty function as $\varphi(s^2) = (s^2 + \epsilon)^{0.5}$, $\epsilon = 0.001$, which is different from the one we used for the data term in Eq. (3.2). Although the Lorentzian penalty provides the sharper motion boundaries [126], the Charbonnier penalty is convex which yields easier optimisation and faster converge. Such mixture regularisations (Lorentzian for data term and Charbonnier for regularisation term) is reported to give fast speed and good robustness to outliers. More analysis for regularisation can be found in Sec. 2.1.3 and Tab. 2-1.

3.2.2 Discrete Laplacian Mesh Energy

In order to improve optical flow estimation against the local complexity of nonrigid motion, a novel *Laplacian Mesh Energy* concept is proposed in this section. The aim of this energy is to account for nonrigid motion in scene deformation. This concept is

inspired by *Laplacian Mesh Deformation* research in graphics, which aims to preserve local mesh smoothness under non-linear transformation [122]. The usage of this concept in computer vision research for optical flow estimation is introduced for the first time here. Although nonrigid motion is highly nonlinear, the pixel movements in such deformations often exhibits strong correlations in local regions. To represent this, we propose a quantitative *Mesh Deformation Weight* based on Laplacian coordinates. The scheme was originally presented by Meyer *et al.* [90] for mesh deformation. Let $\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ be a triangular mesh where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ describes geometric positions of the vertices in absolute cartesian coordinates, $\mathbf{E}$ denotes the set of edges, and $\mathbf{F}$ the set of faces. Considering a small mesh region, each vertex v_i has a *neighbourhood ring* denoted by $\mathcal{N}_i = \{j \mid (i,j) \in \mathbf{E}\}$ which is the set of adjacent vertices of vertex v_i. The *degree* d_i of v_i is the number of elements in $\mathcal{N}_i$. Here the mesh geometric motion is described by differentials instead of absolute Cartesian coordinates. We define the differentials set as $\mathbf{L} = \{\delta_1, \delta_2, \ldots \delta_n\}$ where the coordinate is presented as the difference between the vertex v_i and the geometric average of its neighbours, i.e. $\delta_i = \mathcal{L}(v_i)$. We have

$$\mathcal{L}(v_i) = v_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} v_j. \tag{3.4}$$

These uniform weights are found sufficient for the 2D mesh in our evaluation. Next, we have the mesh energy in Laplacian coordinates as follows:

$$E_{Lap}(\mathbf{w}) = \sum_{i=1}^{n} \|\mathcal{L}(v_i + \mathbf{w}_i) - \mathcal{L}(v_i)\|^2 \tag{3.5}$$

Where $\mathbf{w}_i$ denotes the motion of the vertices v_i. This term of the energy function penalises the shape variance of *neighbourhood ring* after vertex motion. The rationale of using this energy is that the Laplacian coordinates $\mathbf{L}$ encode relative information between vertices and can therefore be used to preserve shape under mesh deformation. In a similar form, the bending energy is also widely used to simulate the deformation of elastic surfaces [142], which is mathematically invariant under a group of transformations in particular under rigid motions and uniform scaling of the surface. However the bending energy is not suitable for our optical flow framework because it is used to calculate the positions of mechanical equilibrium on the 3D deformable object and it is highly computational consuming. In the next section, we will give more details of our optical flow framework.

Input: two images I_1 and I_2
1. Edge-Aware Mesh $\mathcal{M}_1$ Initialization (Sec. 3.3.1)
2. n-levels Gaussian pyramids are constructed for both the images and the mesh. Set the initial pyramid level $k = 0$ and initial flow field $\mathbf{w}^k = (0, 0)^T$
3. The flow field is propagated to level $k + 1$
  4. Detail-Aware Flow Field Enhancement (Sec. 3.3.2)
    4.1 Estimate the tracked mesh $\mathcal{M}_2^k$ for I_2^k.
    4.2 Flow field Enhancement using $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$.
  5. Hybrid Energy optimisation (Sec. 3.3.3)
    5.1 Generate continuous *Laplacian Mesh Energy* using meshes $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$.
    5.2 Nested fixed point iterations.
6. If $k \neq n - 1$ then $k = k + 1$ and go to step 3
Output: optical flow field

Table 3.1: The overall framework of our optical flow model.

3.3 Optical Flow Framework

Tab. 3.1 outlines our overall optical flow framework. In order to utilise the *Laplacian Mesh Energy* it is required to create a mesh over the initial image I_1. Ideally, we desire that the triangles of this mesh do not overlap boundaries in the scene as this may lead to distortions given parallax motion between objects at different depths. We therefore first present an *Edge-Aware Mesh Initialization* scheme (Sec 3.3.1) as part of our framework.

We also present a novel coarse-to-fine pyramidal framework [20] to utilise our *Laplacian Mesh Energy* in a variational model. In our framework we overcome a previous limitation of such pyramidal approaches, i.e. the loss of small flow details when propagating flow field from coarse to finer pyramidal levels. In such cases, small image details at a finer level of the pyramid are lost due to flow computation being initially performed on a coarsely sampled version of the image. As such, the flow for these detailed regions is not remained and propagated to the finer level.

Finally, an optimisation scheme (Sec. 3.3.3) is proposed to minimise the discrete *Laplacian Mesh Energy* on every level of the pyramidal framework. In the following sub-sections each step is described in detail.

3.3.1 Edge-Aware Mesh Initialization

The proposed algorithm is input by an image pair and a mesh with triangle edges that follow object boundaries in one of the images as closely as possible. We will discuss the implications of mesh design and its effect on our algorithm behaviour in the evaluation. The underlying mesh is an essential part of *Laplacian Mesh Energy* computation. Us-

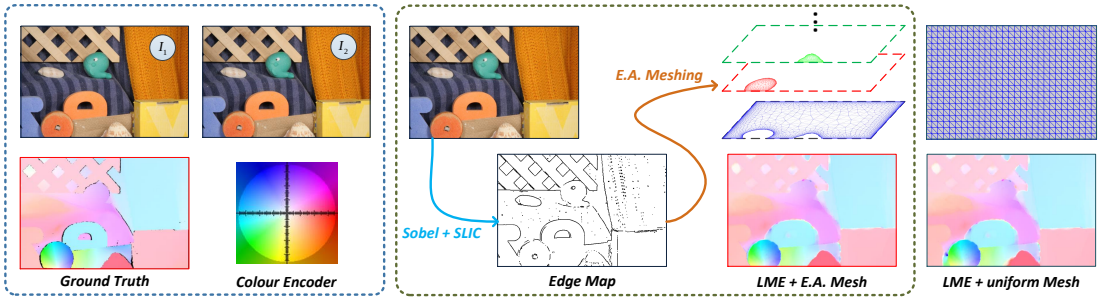


Figure 3-1: Edge-aware mesh initialisation process on a sample sequence *RubberWhale* [5].

ing a uniform mesh (Fig. 3-1) with equal distances between vertices along its horizontal and vertical adjacent neighbours is one strategy that can be employed in our approach. However, in such a case the grid elements within the mesh will typically overlap the boundaries of objects scene, which results in unexpected errors in our energy minimisation. This is because triangles within the mesh will be skewed given parallax motion between different objects at different image depths, resulting in a noisier flow field in these areas.

In order to address this issue, we propose an edge-aware meshing scheme in Fig. 3-1, which operates as follows: First, we create two edge maps on the input image using *SLIC Superpixels* [1] and *Sobel Kernel* edge detection respectively. We then apply a binary *AND Operation* on the two edge maps in order to deduce uncommon edges, and remove noise using a Gaussian filter. The rationale behind this approach is that the *Sobel kernel* returns a large number of candidate edges, but also multiple false-positive noise like edges relating to image detail as opposed to object boundaries. The *SLIC Superpixels* on the other hand is less likely to create boundaries relating to image detail. Performing an AND operation eliminates a great deal of the noisy edge boundaries and retains a large proportion of reliable ones. Finally, we construct a triangular mesh $\mathcal{M}_1$ using *Delaunay* triangulation on the points of the edge map.

Given the input mesh $\mathcal{M}_1$, an n-level image pyramid is built (Tab. 3.1). The input images I_1, I_2 along with the mesh $\mathcal{M}_1$ are resized with the same sampling rate on each level, denoted by I_1^k, I_2^k and $\mathcal{M}_1^k$, where $k = 1, 2, \ldots, n$. We then perform *Detail-Aware Flow Field Enhancement* and *Hybrid Energy optimisation* on each level.

3.3.2 Detail-Aware Flow Field Enhancement

As mentioned in the beginning of Sec. 3.3, the aim of this step is to preserve small flow details which may be lost when propagated from the adjacent coarser level. First, we estimate a tracked mesh $\mathcal{M}_2^k$ on I_2^k by propagating the mesh $\mathcal{M}_1^k$ from I_1^k onto I_2^k. Next, we build a labelling model using vertex displacement vectors from $\mathcal{M}_1^k$ to $\mathcal{M}_2^k$ and the flow field from coarser level. This labelling model is then solved to retain small flow details. The whole process is detailed in the next two sections.

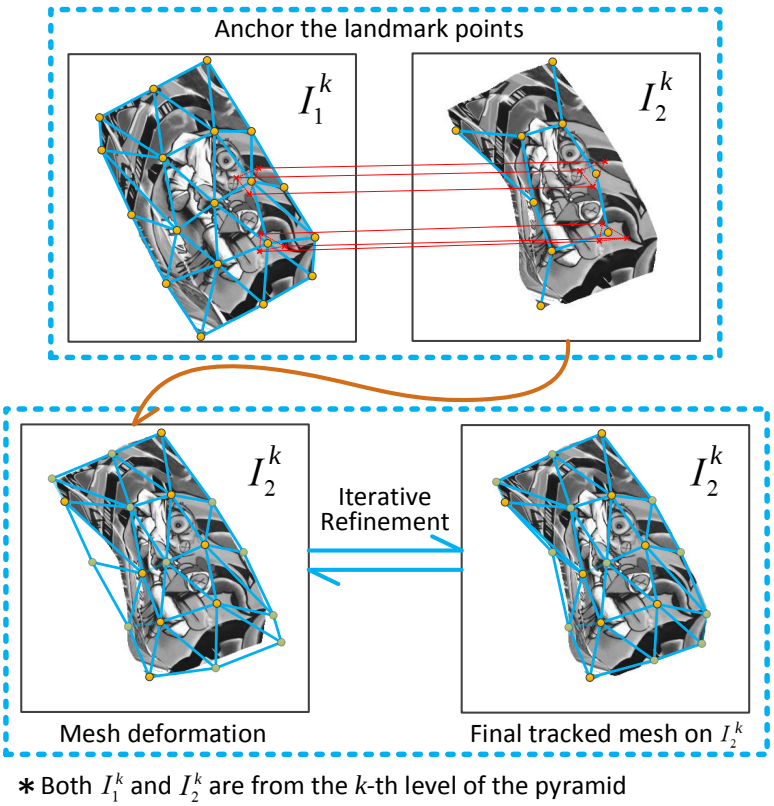


Figure 3-2: Frame-frame tracked mesh $\mathcal{M}_2^k$ estimation process on the k-th level of the coarse-to-fine framework.

Frame-to-Frame Tracked Mesh $\mathcal{M}_2$ Estimation

In order to propagate the mesh from $\mathcal{M}_1^k$ to $\mathcal{M}_2^k$ at pyramid level k, we employ an SIFT-based *Anchor Patch* technique and *Laplacian Mesh Deformation*, which utilises I_1^k, I_2^k and $\mathcal{M}_1^k$. As shown in Fig. 3-2, we follow the *Anchor Patch* process outlined in Sec. 5.5 to achieve this mesh propagation: SIFT features are initially detected and matched between images I_1^k and I_2^k. We then go through every vertex v of $\mathcal{M}_1^k$ and search for the three nearest SIFT features f_* within a 9×9 search window centred on the vertex v in I_1^k. As shown in Fig 3-2 (**Top Row**), the corresponding features in I_2^k and *Barycentric Coordinate Mappings* – defined by the triangle form of the 3 SIFT features – are used to calculate a correspondent vertex v' for $\mathcal{M}_2^k$ in I_2^k.

Next, we apply an error function $Err(v \rightarrow v')$ on all the newly created vertex correspondences between I_1^k and I_2^k, where $v \rightarrow v'$ is the matching between vertices v and v'; v' denotes the correspondent vertex in I_2^k. Given location of the vertex v is $(x, y)^T$ in Cartesian Coordinate, a displacement vector from vertex v to v' is denoted by $(u, v)^T = v' - v$. The *Error Score* $Err(v \rightarrow v')$ is calculated as the weighted *Root Mean Square* (RMS) error at a 3×3 pixel area centred on locations $(x, y)^T$ and $(x+u, y+v)^T$ in images I_1^k and I_2^k respectively.

$$
\begin{aligned}
Err(v \rightarrow v') &= \sqrt{\frac{\alpha_1 d(x,y) + \alpha_2 d_{cross}(x,y) + \alpha_3 d_{diag}(x,y)}{\alpha_1 + \alpha_2 + \alpha_3}} \\
d_{diag}(x,y) &= d(x-1,y-1) + d(x+1,y+1) \\
&+ d(x-1,y+1) + d(x+1,y-1) \\
d_{cross}(x,y) &= d(x-1,y) + d(x+1,y) + d(x,y-1) + d(x,y+1) \\
d(x,y) &= |I_1^k(x,y) - I_2^k(x+u,y+v)|^2
\end{aligned} \tag{3.6}
$$

Where α_1, α_2 and α_3 are weights for controlling the contribution of each pixel in the 3×3 window. In our experiments, all these weights are set as $\alpha_1 = 1$, $\alpha_2 = 0.25$ and $\alpha_3 = 0.125$. This is carried out in order to select the most reliable vertex matches between the two images. If the $Err(v \rightarrow v')$ results in low errors, the vertices v and v' are selected as sets of *Control Points* – defined here as $\mathbf{V}_c, \mathbf{V}'_c$ – Note that set $\mathbf{V}_c$ contains the control points v from $\mathcal{M}_1^k$ while set $\mathbf{V}'_c$ contains the control points v' from $\mathcal{M}_2^k$. Thus we have the formal form $\{\mathbf{V}_c, \mathbf{V}'_c | \forall v_{c\cdot i} \in \mathbf{V}_c, \forall v'_{c\cdot i} \in \mathbf{V}'_c, Err(v_{c\cdot i} \rightarrow v'_{c\cdot i}) < \eta\}$, η is our predefined error threshold. This method of creating/selecting control points between meshes has been utilised to obtain reliable anchor patches between images in our dense long term tracking framework (Sec. 5.5).

Finally, in order to estimate the positions of the remaining vertices in $\mathcal{M}_2^k$, *Laplacian Mesh Deformation* [122] is applied using $\mathcal{M}_1^k$ and the corresponding control points $\mathbf{V}_c$ and $\mathbf{V}'_c$. We minimise the following function to achieve this:

$$
\min_{\mathbf{V}'} \{\|\mathbf{L}\mathbf{V}' - \mathbf{L}\mathbf{V}\|^2 + \sum_{i=1}^{m} \|v'_{c\cdot i} - v_{c\cdot i}\|^2\} \tag{3.7}
$$

where $\mathbf{L}$ is a Laplacian matrix computed using Eq. (3.4), $\mathbf{V}$ represents the vertex set of $\mathcal{M}_1^k$. $\mathbf{V}_c$ and $\mathbf{V}'_c$ are control points set where we have $v_c \in \mathbf{V}_c, v'_c \in \mathbf{V}'_c, v_c \rightarrow v'_c$ and m is the number of control points. After minimising [122] Eq. (3.7), we obtain our initial mesh $\mathcal{M}_2^k$ for I_2^k, and denote this set of vertices as $\mathbf{V}'$.

However, this set of vertices $\mathbf{V}'$ may contain outliers due to the limited number of control points in $\mathbf{V}_c$ and $\mathbf{V}'_c$. We therefore propose an iterative refinement algorithm as shown in Tab. 3.2 and Fig. 3-2 (**Bottom Row**). In each iteration of our algorithm, we apply the evaluation function (Err) on the matches between $\mathbf{V}$ and $\mathbf{V}'$ to obtain low-scoring matches by which the control point sets $\mathbf{V}_c$ and $\mathbf{V}'_c$ are updated. The updated $\mathbf{V}_c$ and $\mathbf{V}'_c$ are then propagated onto the next iteration until all the matches between $\mathbf{V}$ and $\mathbf{V}'$ reach the *Error Score* threshold (under η). In our implementation, $\mathbf{V}_c = \mathbf{V}$ normally converges within 15 iterations.

Algorithm 1: Iterative Refinement Algorithm

$\mathbf{V}, \mathbf{V}', \mathbf{V}_c, \mathbf{V}'_c$
$\mathbf{V}_c \subset \mathbf{V}, \mathbf{V}'_c \subset \mathbf{V}'$
$\mathbf{V}_c \rightarrow \mathbf{V}'_c$
while not $\mathbf{V}_c = \mathbf{V}$
    $\mathbf{V}' := \underset{\mathbf{V}'}{min}\{\|\mathbf{L}\mathbf{V}' - \mathbf{L}\mathbf{V}\|^2 + \sum_{i=1}^{m} \|v'_{c \cdot i} - v_{c \cdot i}\|^2\}$
    for all $v \in \mathbf{V}, v' \in \mathbf{V}'$ **do**
        if $Err(v \rightarrow v') < \eta$ **then**
            $\mathbf{V}_c := \mathbf{V}_c \cup \{v\}, \mathbf{V}'_c := \mathbf{V}'_c \cup \{v'\}$
        end if
    end for
end while

Table 3.2: The iterative refinement algorithm for tracked mesh $\mathcal{M}_2^k$ estimation.

Inter-Level Flow Field Enhancement

In this section we consider the small flow details encoded by mesh $\mathcal{M}_2^k$ from the previous section. In existing pyramidal approaches, small motion displacements on a finer level can be lost when the flow field is propagated from a coarser level. To address this issue we utilise a inter-level labelling model which identifies discrepancies between the propagated flow field $\mathbf{w}$ and displacement vectors $\mathbf{w}'$ of **mesh vertices**. Given vertex sets $\mathbf{V}$ of mesh $\mathcal{M}_1^k$ for image I_1^k, $\mathbf{V}'$ of mesh $\mathcal{M}_2^k$ for image I_2^k, $\mathbf{w}'$ is defined as a ***sparse*** vector set $\mathbf{w}' = \{w'_1, w'_2, \ldots, w'_n\}$ containing displacement vectors between $\mathbf{V}$ and $\mathbf{V}'$ – n the is number of the vertices – is computed by $w'_i = v'_i - v_i$, where $v_i \in \mathbf{V}, v'_i \in \mathbf{V}'$.

Therefore the enhancement process is outlined in Fig. 3-3. For each w'_i, we consider the flow vector w_i in $\mathbf{w}$ where w'_i and w_i share the same pixel position. First of all, we identify the difference between w'_i and w_i. We compute the Euclidean distance between endpoints of w'_i and w_i, and for extra robustness, we also compute the Euclidean distances between endpoints of w'_i and the 8 adjacent neighbours (within a 3×3 window) of w_i. The displacement w'_i is regarded as a potential flow candidate only if all 9 Euclidean distances are larger than 1 pixel. This selection strategy can keep a reasonable number of the candidates that are most informatic. This identification operation is repeated on all n displacement vectors in $\mathbf{w}'$ to give a new flow candidate list $\{w'_{g \cdot 1}, w'_{g \cdot 2}, \ldots, w'_{g \cdot m}\}$ where $m \leqslant n$. For HD images in our experiments, the number of new flow candidates is at most 25 on the finest level and even significantly less on the coarser levels (For the Middlebury images, the number is between 7 and 14; and it is between 5 and 10 for the *Original*, Garg dataset; this number is between 20 to 25 for Sintel images, see Sec.3.4). These new flow candidates are typically distributed widely over the whole image including generally featureless regions (as opposed to SIFT [152] feature matching).

Having obtained the new flow candidate list, we build a labelling model. We assume

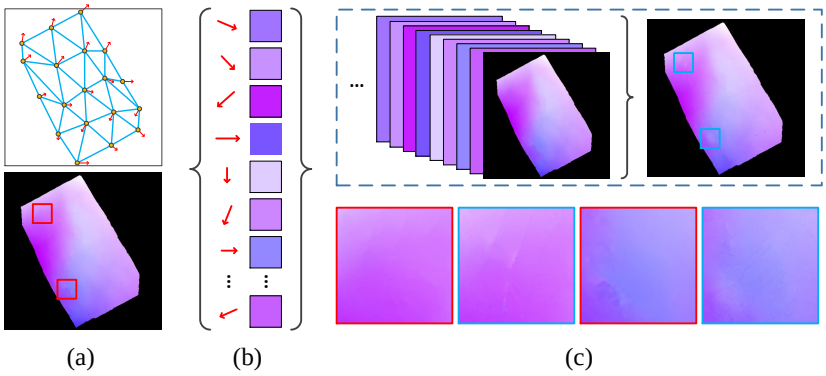


Figure 3-3: Small flow Details Preservation. (a) **Top**: The mesh and vertex displacement vectors (red arrows). **Bottom**: The flow field $\mathbf{w}$ propagated from the adjacent coarser level. (b) Flow candidates: the selected vertex displacement vectors (red arrows) and the flow vectors (colour coding) at the same pixel location. (c) **Top**: The labelling model optimised using QPBO. **Bottom**: The visual comparison of closeups between $\mathbf{w}$ (red) and the optimised flow field $\widehat{\mathbf{w}}$ (blue).

that each pixel of the image (on specific level) has $m+1$ labels to be selected from either the new flow candidates $\{w'_{g\cdot 1}, w'_{g\cdot 2}, \ldots, w'_{g\cdot m}\}$ or the original flow vector. We also adopt the endpoint distance as the pairwise cost between labels. *Quadratic Pseudo-Boolean optimisation* (QPBO) is then employed to solve this problem. Considering the computation, we apply a fast QPBO implementation [106] to handle the multi-labels model [69] which has previously been used in discrete optical flow methods for optimisation [152]. In this work the flow candidates can potentially retain smaller flow details that would otherwise be lost in pyramidal flow propagation, or feature matching that might be less robust and more sparser given a textureless surface.

The process described above (Fig. 3-3) outputs a flow field $\widehat{\mathbf{w}}^k$ and $\widehat{\mathbf{w}}^k = (\widehat{u}^k, \widehat{v}^k)^T$, which is then used as the initial flow field for computing $\mathbf{w}^{k+1}$ on level $k+1$ as below.

3.3.3 Hybrid Energy optimisation

Due to the highly non-linear nature of the energy function $E(\mathbf{w})$, its optimisation is an essential part of our algorithm. In this section, we introduce a numerical scheme to minimise this hybrid energy w.r.t. the discrete Laplacian mesh energy and the continuous brightness energy. We initially define mathematical abbreviations (similar to [20]) for our brightness energy minimisation as follows:

$$\begin{aligned} I_x &= \partial_x I_2(\mathbf{x}+\mathbf{w}) & I_{yy} &= \partial_{yy} I_2(\mathbf{x}+\mathbf{w}) \\ I_y &= \partial_y I_2(\mathbf{x}+\mathbf{w}) & I_{xx} &= \partial_{xx} I_2(\mathbf{x}+\mathbf{w}) \\ I_z &= I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x}) & I_{xz} &= \partial_x I_2(\mathbf{x}+\mathbf{w}) - \partial_x I_1(\mathbf{x}) \\ I_{xy} &= \partial_{xy} I_2(\mathbf{x}+\mathbf{w}) & I_{yz} &= \partial_y I_2(\mathbf{x}+\mathbf{w}) - \partial_y I_1(\mathbf{x}) \end{aligned} \tag{3.8}$$

In order to minimise the mesh energy in our variational model, we define its uniform weights in polar coordinates. We have $\mathcal{L} = (\mathcal{L}_r, \mathcal{L}_\theta)^T$ where $\mathcal{L}_r$ denotes the magnitude component and $\mathcal{L}_\theta$ denotes the angle component, which results in two terms for the Laplacian mesh energy as follows:

$$\begin{aligned} E_{Lap}(\mathbf{w}) = \lambda \int_\Omega \psi(\|\mathcal{L}_{r\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{r\cdot 1}(\mathbf{x})\|^2) d\mathbf{x} \\ +\lambda \int_\Omega \psi(\|\mathcal{L}_{\theta\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{\theta\cdot 1}(\mathbf{x})\|^2) d\mathbf{x} \end{aligned} \tag{3.9}$$

where both the terms $\mathcal{L}_{*\cdot 1}$ and $\mathcal{L}_{*\cdot 2}$ are computed respectively based on $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$ (Sec. 3.3.2) on level k. Note that the terms are applied to each pixel of the input image. We go through each triangle and employ bicubic interpolation using the $\mathcal{L}_*$ values on the three vertices of the triangle. This process results in a continuous *Laplacian Mesh Energy* presented in Eq. (3.9). The term λ is a weight capturing the influence of our Laplacian mesh, and is set to 0.6 in our experiments. The behaviour of our algorithm by varying λ values is also considered in the evaluation section. The mathematical abbreviations for the *Laplacian Mesh Energy* are as follows:

$$\begin{aligned} \mathcal{L}_{*\cdot x} &= \partial_x \mathcal{L}_*(\mathbf{x}+\mathbf{w}) \\ \mathcal{L}_{*\cdot y} &= \partial_y \mathcal{L}_*(\mathbf{x}+\mathbf{w}) \\ \mathcal{L}_{*\cdot z} &= \mathcal{L}_{*\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{*\cdot 1}(\mathbf{x}) \end{aligned} \tag{3.10}$$

Our energy function $\mathbf{E}(\mathbf{w})$ is highly nonlinear on the terms of $\mathcal{L}_*$, $\mathbf{w}$ and ψ. We employ two nested *Fixed Point Iterations* on $\mathbf{w}$ after Euler-Lagrange equations are applied.

Fix w for I_*^{k+1} and $\mathcal{L}_*^{k+1}$. In the first fixed point iteration, the algorithm goes through every level of the pyramid starting from the top/coarsest level. We assume that $\mathbf{w}$ converges at the k-th iteration (the k-th level of the pyramid) giving us $\mathbf{w}^k = (\mathbf{u}^k, \mathbf{v}^k)^T, k = 0, 1, \ldots$ with an initialization $\mathbf{w}^0 = (0,0)^T$ at the coarsest level of the pyramid. The flow field $\mathbf{w}^k$ is then propagated to the next finer level for computing the initial flow field $\widehat{\mathbf{w}}^k$ (sec. 3.3.2). However, the new system reached fixed $\mathbf{w}^k$ is still nonlinear and difficult to solve as it contains terms I_*^{k+1}, $\mathcal{L}_*^{k+1}$ and the nonlinear function ψ'.

Fix $d\mathbf{w}$ for ψ'. First order Taylor expansion is employed on the terms I_z^{k+1}, I_{xz}^{k+1}, I_{yz}^{k+1}, $\mathcal{L}_{*\cdot x}^{k+1}$, $\mathcal{L}_{*\cdot y}^{k+1}$ and $\mathcal{L}_{*\cdot z}^{k+1}$ in order to remove the nonlinearity of I_*^{k+1} and $\mathcal{L}_*^{k+1}$. We have $I_z^{k+1} \approx I_z^k + I_x^k du^k + I_y^k dv^k$ and $\mathcal{L}_{*\cdot z}^{k+1} \approx \mathcal{L}_{*\cdot z}^k + \mathcal{L}_{*\cdot x}^k du^k + \mathcal{L}_{*\cdot y}^k dv^k$, where we assume that the flow field on level $k+1$ can be estimated by the flow field and the incremental

from previous level k, denoted as $\mathbf{w}^{k+1} \approx \widehat{\mathbf{w}}^k + d\mathbf{w}^k$. Two unknown increments du^k, dv^k and two known flow fields $\widehat{u}^k$,$\widehat{v}^k$ can be obtained from the previous iteration. Note that this assumption also applies to the terms $\mathcal{L}^{k+1}_{*\cdot z}$. For removing nonlinearity in ψ' with unknown increments du^k and dv^k, we apply a nested second fixed point iteration. Here, in every iteration step we assume that both $du^{k,j}$ and $dv^{k,j}$ converge within j iteration steps with initialization of $du^{k,0} = 0$ and $dv^{k,0} = 0$. Therefore, the final linear system is obtained in $du^{k,j+1}$ and $dv^{k,j+1}$ as follows:

$$\begin{aligned}
&(\psi')^{k,j}_{Data} \cdot (I^k_x(I^k_z + I^k_x du^{k,j+1} + I^k_y dv^{k,j+1}) \\
&+\theta\ [I^k_{xx}(I^k_{xz} + I^k_{xx} du^{k,j+1} + I^k_{xy} dv^{k,j+1}) \\
&+I^k_{xy}(I^k_{yz} + I^k_{xy} du^{k,j+1} + I^k_{yy} dv^{k,j+1})]) \\
&+\lambda\ (\psi')^{k,j}_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot x}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^{k,j+1} + \mathcal{L}^k_{r\cdot y} dv^{k,j+1}) \\
&+\lambda\ (\psi')^{k,j+1}_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot x}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^{k,j+1} + \mathcal{L}^k_{\theta\cdot y} dv^{k,j+1}) \\
&-\xi\ \mathbf{Div}(\varphi')^{k,j}_{Smooth} \cdot \nabla(u^k + du^{k,j+1}) = 0
\end{aligned} \tag{3.11}$$

$$\begin{aligned}
&(\psi')^{k,j}_{Data} \cdot (I^k_y(I^k_z + I^k_x du^{k,j+1} + I^k_y dv^{k,j+1}) \\
&+\theta\ [I^k_{yy}(I^k_{yz} + I^k_{xy} du^{k,j+1} + I^k_{yy} dv^{k,j+1}) \\
&+I^k_{xy}(I^k_{xz} + I^k_{xx} du^{k,j+1} + I^k_{xy} dv^{k,j+1})]) \\
&+\lambda(\psi')^{k,j}_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot y}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^{k,j+1} + \mathcal{L}^k_{r\cdot y} dv^{k,j+1}) \\
&+\lambda(\psi')^{k,j}_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot y}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^{k,j+1} + \mathcal{L}^k_{\theta\cdot y} dv^{k,j+1}) \\
&-\xi\ \mathbf{Div}(\varphi')^{k,j}_{Smooth} \cdot \nabla(v^k + dv^{k,j+1}) = 0
\end{aligned} \tag{3.12}$$

Where $(\psi')^k_{Data}$ and $(\psi')^k_{Lap\cdot *}$ provides both robustness against occlusion and sharpness on object boundaries, $(\varphi')^k_{Smooth}$ is defined as diffusivity in the global smoothness terms [20] as below:

$$\begin{aligned}
(\psi')^k_{Data} &= \psi'((I^k_z + I^k_x du^k + I^k_y dv^k)^2 \\
+\theta[(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)^2 &+ (I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)^2]) \\
(\psi')^k_{Lap\cdot *} &= \psi'(\mathcal{L}^k_{*\cdot z} + \mathcal{L}^k_{*\cdot x} du^k + \mathcal{L}^k_{*\cdot y} dv^k)^2 \\
(\varphi')^k_{Smooth} &= \varphi'(\left\|\nabla(u^k + du^k)\right\|^2 + \left\|\nabla(v^k + dv^k)\right\|^2)
\end{aligned}$$

In our implementation, an n-level image pyramid is constructed by using a down sampling factor of 0.75 and *Bicubic Interpolation* on each pyramid level. Furthermore, the first fixed point iterations are set based on both the down sampling factor and the image size while the nested second fixed point iterations are fixed to 5 steps. Finally, the large linear systems (Eq. (3.11) and (3.12)) are solved using *Conjugate Gradients*

with 45 iterations. More mathematical details of our hybrid energy optimisation can be found in the appendix A.1.

3.4 Evaluation

In this section we evaluate the performance of our approach and compare its performance with existing state-of-the-art techniques. We use quantitative metrics to demonstrate the performance of our approach against the highest performing existing methods on the *Middlebury* dataset [5] and a synthetic benchmark dataset with ground truth introduced by Garg *et al.* [49]. As our method is designed to be particularly suitable for nonrigid scenarios, we therefore compare our approach against a number of the best performing (state-of-the-art) publicly available nonrigid optical flow algorithms, details are as below: The first nonrigid algorithm we have used for comparison is Garg *et al.*'s spatio-temporal optical flow algorithm. Their approach exploits correlations between 2D trajectories of neighbouring pixels to improve optical flow estimation. In addition, We compare our results with the *Improved TV-L1* (ITV-L1) algorithm [143] and Brox *et al.* [20]. The former has a similar optimisation framework and preprocessing steps to that of Garg *et al.* and ranks in the reasonable midfield of the *Middlebury* evaluation based on overall average. The latter is proposed by Brox *et al.* to overcome the issues caused by large pixel displacements with the help of integrating the image pyramid and warping technique in a variational model. Finally, We compare our method with the state-of-the-art keypoint-based nonrigid image registration method proposed by Pizarro *et al.* [100]. Note that all experiments are performed using a 2.9Ghz Xeon 8-cores, NVIDIA Quadro FX 580, 16Gb memory computer.

In summary, our results show that the *Laplacian Mesh Energy* greatly improves algorithm performance while our algorithm outperforms all publicly available nonrigid optical flow techniques. It also performs in the top tier of all the *Middlebury* criteria, and strongly overall - especially compared to the aforementioned specialist nonrigid optical flow techniques.

3.4.1 Middlebury Dataset

We first performed an evaluation on the *Middlebury* benchmark dataset using default parameter setting as follows: $\theta = 0.6$, $\lambda = 0.6$ and $\xi = 0.75$ are set for the energy function $E(\mathbf{w})$ while $\eta = 0.25$ is applied in vector displacement candidate selection (Sec. 3.3.2). These parameter setting remains consistent in all experiments in this chapter.

As shown in Fig. 3-4, Our implementations is denoted by *LME* with the automatic *Edge-Aware Mesh Initialization* (Sec. 3.3.1). We observe that *LME* ranks among

Average endpoint error	avg.	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [74]	4.2	0.08 4	0.21 2	0.07 9	**0.15** 1	**0.48** 1	**0.11** 1	**0.20** 1	**0.40** 1	**0.14** 1	0.15 8	0.80 15	0.08 6	0.63 6	0.93 6	0.43 6	**0.26** 1	**0.76** 1	0.23 3	0.11 9	0.12 6	0.17 11	**0.38** 1	**0.79** 1	**0.44** 1
ADF [71]	9.5	0.08 4	0.22 6	**0.06** 1	0.18 4	0.62 9	0.14 9	0.29 14	0.71 18	0.17 8	0.16 17	0.91 28	**0.07** 1	0.69 13	1.03 13	0.47 8	0.43 13	0.91 4	0.28 6	0.12 15	0.12 6	0.20 20	0.43 2	0.88 2	0.63 7
LME [76]	9.7	0.08 4	0.22 6	**0.06** 1	**0.15** 1	0.49 2	**0.11** 1	0.30 17	0.64 9	0.31 51	0.15 8	0.78 13	0.09 13	0.66 8	0.96 8	0.53 16	0.33 3	1.18 17	0.28 6	0.12 15	0.12 6	0.18 14	0.44 3	0.91 5	0.61 5
IROF++ [61]	9.8	0.08 4	0.23 9	0.07 9	0.21 18	0.68 16	0.17 19	0.28 10	0.63 8	0.19 18	0.15 8	0.73 10	0.09 13	0.60 3	0.89 4	0.42 5	0.43 13	1.08 11	0.31 14	0.10 4	0.12 6	0.12 4	0.47 7	0.98 9	0.68 13
⋮	⋮		⋮			⋮						⋮						⋮						⋮	
TV-L1-improved [17]	37.6	0.09 17	0.26 25	0.07 9	0.20 15	0.71 20	0.16 16	0.53 40	1.18 46	0.22 34	0.21 39	1.24 47	0.11 29	0.90 37	1.31 42	0.72 31	1.51 61	1.93 57	0.84 57	0.18 56	0.17 56	0.31 58	0.73 36	1.62 41	0.87 33
⋮	⋮		⋮			⋮						⋮						⋮						⋮	
Brox et al. [5]	40.5	0.11 33	0.32 43	0.11 49	0.27 42	0.93 43	0.22 45	0.39 30	0.94 29	0.24 38	0.24 41	1.25 48	0.13 36	1.10 59	1.39 54	1.43 66	0.89 47	1.77 52	0.55 44	0.10 4	0.13 20	0.11 2	0.91 47	1.83 51	1.13 49

Figure 3-4: Snapshot of *Average Endpoint Error* (AEE) in *Middlebury* Evaluation (Captured on October 2^{nd}, 2012). Our proposed method is *LME* with automatic Edge-Aware mesh initialization. The average computational time is recorded as 476 seconds.

the top three algorithms and significantly outperforms most methods in the *Average Endpoint Error* (AEE) test with an overall average rank 9.7. Moreover, Fig. 3-5 shows the visual comparison of both our implementations (*LME* and *LME-Manual*) on the Middlebury dataset [5]. Note that *LME* uses an automatic *Edge-Aware Mesh Initialization* while *LME-Manual* takes a manually segmented mesh as input. The former is a completely unsupervised algorithm while the latter gives more accurate and sharper flow details on the object boundaries because the manual segmentation fully overcomes the problem of the unrealistic mesh deformation on the object boundaries.

However, *Middlebury* results against other nonrigid approaches (Garg *et al.*'s and Pizarro *et al.*'s methods) are not available. We therefore compare our approach against theirs using a specific nonrigid ground truth dataset (Sec. 3.4.2). Our approach also ranks in the top three overall for the *Average Normalized Interpolation Error* (ANIE) test which represents the quality of local image detail during the warping. Particularly strong performance is observed on *Middlebury* sequences captured using the high-speed camera – *Backyard*, *Basketball*, *Dumptruck* and *Evergreen*.

Fig. 3-6 shows an experiment to evaluate the effect by varying the number of input features. We randomly select different numbers (from 0% to 100%) of features from the initial full detection feature set before performing *Frame-to-Frame Tracked Mesh* $\mathcal{M}_2$ *Estimation* step (Sec.3.3.2). Here 0% denotes that no feature is taken into account. In this case, the *Discrete Laplacian Mesh Energy* is turned off ($\lambda = 0$, in the main energy Eq. (3.1)) because control points cannot be detected. 100% present a full set of detected features. For better observation, we show the full set (5706 in total) of SIFT features on the input image; And we calculate the AEE on two selected patches (A and B, Fig. 3-6 top-right). The former is from the background surface while the latter is picked around the object boundary. It can be observed that, in the general case (*Dimetrodon* in Fig. 3-6 top-left), a very small amount of features (10%) leads to slightly higher AEE against the case of turning off the *Discrete Laplacian Mesh Energy* (0%). We believe that 10% features cannot provide enough information for control points calculation which results in a wrong mesh estimation. Apart from this, more features involved yields lower error of optical flow estimation on both single object surface (Patch A) and object boundary (Patch B).

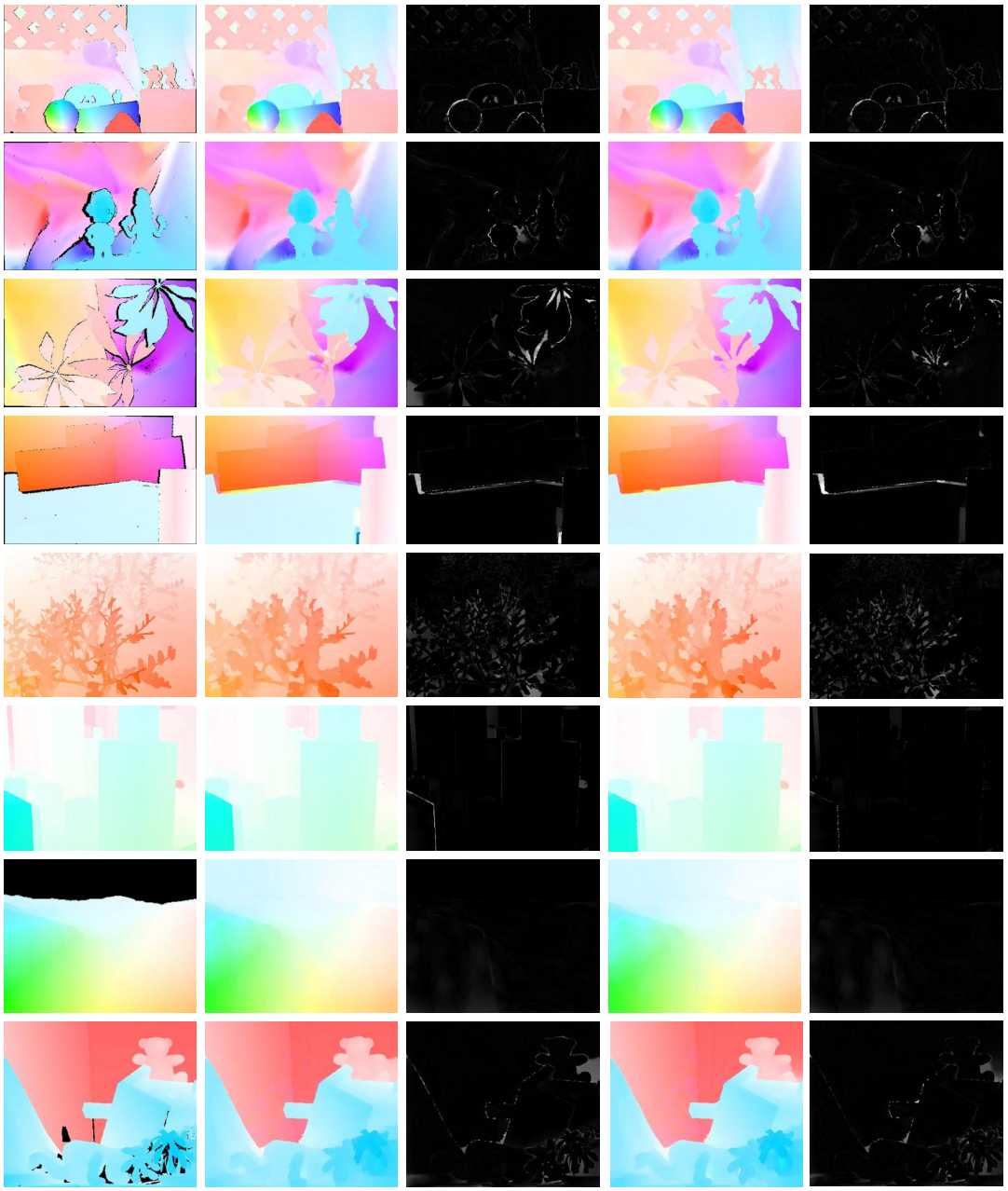


Figure 3-5: The Visual Comparison on Middlebury Dataset [5]. (a): The ground truth flow fields. (b) and (c): *LME* results and the error maps. (d) and (e): *LME-Manual* results and the error maps. **Rows from top to bottom**: The sequences *Army*, *Mequon*, *Schefflera*, *Wooden*, *Grove*, *Urban*, *Yosemite* and *Teddy*.

3.4.2 MOCAP Benchmark Dataset

In this section we compare against a recently popular optical flow dataset specifically designed for nonrigid evaluation in long term. In order to quantitatively evaluate their optical flow algorithm, Garg *et al.* proposed benchmark sequences accompanying with

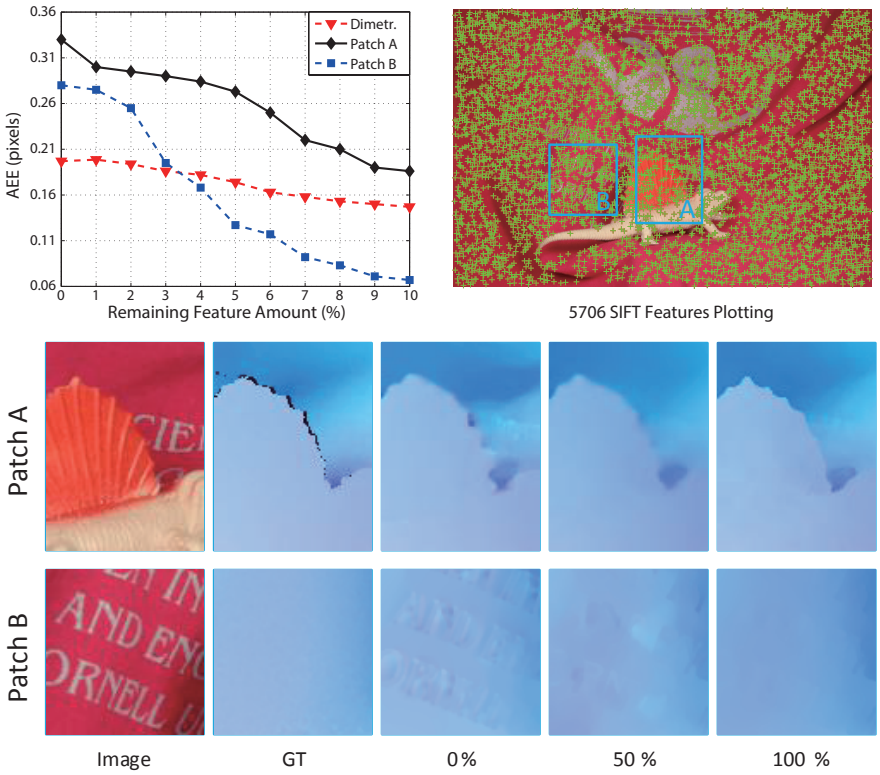


Figure 3-6: AEE measures of *LME* on sequence *Dimetrodon* by varying the number of input features by percentage.

ground truth [49]. A continuous dense 3D surface is obtained by interpolating sparse motion capture (MOCAP) data from real deformations of a waving flag [148]. They then project the dense textured 3D surface synthetically onto the image plane resulting in a sequence of 60 images (500×500 pixel dimension) along with optical flow ground truth motion. Our evaluation is performed on both the original captured sequence and three other degraded sequences from the Garg *et al.* benchmark dataset, which includes: **Synthetic occlusions** – Two black dots with radius of 20 pixels orbit the deformable object. **Gaussian noise** – Added with standard deviation of 0.2 relative to the range of image gray value intensities. **Salt & pepper noise** – Added with a density of 10%.

In this experiment, we calculate the optical flow field from the reference frame to each of remaining frame. The result optical flow fields are then compared to the ground truth. When comparing against the other methods, we use the same parameters cited by other authors. That is, for both Garg *et al.* and ITV-L1, the weights α and β are set to 30 and 2 respectively; and we also use 5 warp iterations and 20 alternation iterations [143]. According to parameter setting in [49], *Principal Components Analysis* (PCA) and *Discrete Cosine Transform* (DCT) are used for the 2D trajectory motion basis of Garg *et al.*. In addition, Brox *et al.* [20] is applied with their default parameter

Methods	Average Endpoint Error (AEE)			
	Original	Occlusion	Guass.N	S&P.N
LME (Ours, Auto-meshing)	**0.39**	**0.65**	1.20	**0.87**
Garg *et al.*, PCA [49]	0.58	0.70	1.62	1.20
Garg *et al.*, DCT [49]	0.57	0.73	1.85	1.52
Pizarro *et al.* [100]	0.76	0.78	**0.95**	0.95
ITV-L1 [143]	0.56	0.69	1.81	1.37
Brox *et al.* [20]	12.62	13.55	13.73	13.32

(a) Average Endpoint Error (AEE) comparison of different methods on Garg *et al.* benchmark dataset [49].

Methods	R 1.0 Endpoint Error (R 1.0)			
	Original	Occlusion	Guass.N	S&P.N
LME (Ours, Auto-meshing)	**0.04**	**0.06**	**0.24**	**0.19**
Garg *et al.*, PCA [49]	0.12	0.16	0.61	0.41
Garg *et al.*, DCT [49]	0.11	0.14	0.68	0.52
Pizarro *et al.* [100]	0.2	0.21	0.24	0.24
ITV-L1 [143]	0.09	0.11	0.68	0.45
Brox *et al.* [20]	0.28	0.32	0.72	0.69

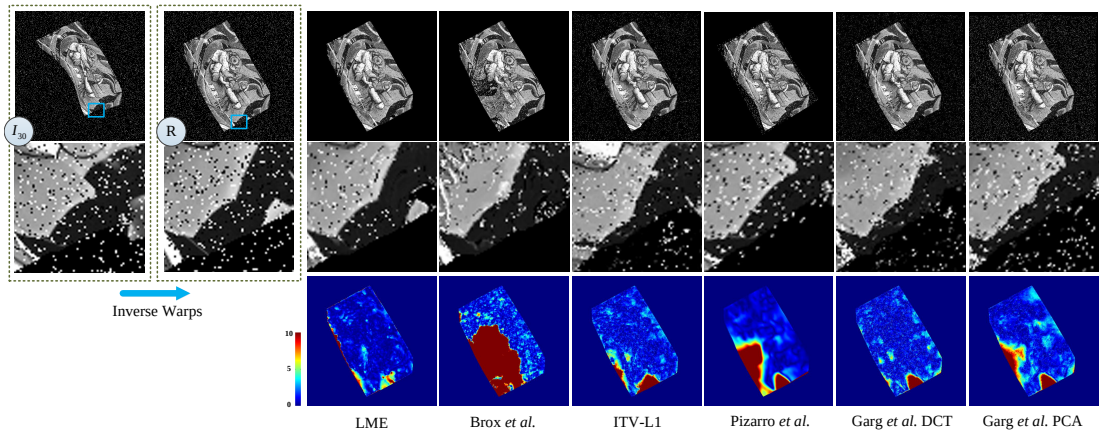
(b) Robustness R 1.0 Endpoint Error (R 1.0) comparison of different methods on Garg *et al.* benchmark dataset [49].

Methods	A 75 Endpoint Error (A 75)			
	Original	Occlusion	Guass.N	S&P.N
LME (Ours, Auto-meshing)	**0.37**	**0.39**	**0.97**	**0.83**
Garg *et al.*, PCA [49]	0.69	0.77	1.98	1.42
Garg *et al.*, DCT [49]	0.63	0.69	2.19	1.81
Pizarro *et al.* [100]	0.88	0.91	0.97	0.97
ITV-L1 [143]	0.50	0.53	2.23	1.58
Brox *et al.* [20]	1.83	9.38	4.99	4.52

(c) Robustness A 75 Endpoint Error (A 75) comparison of different methods on Garg *et al.* benchmark dataset [49].

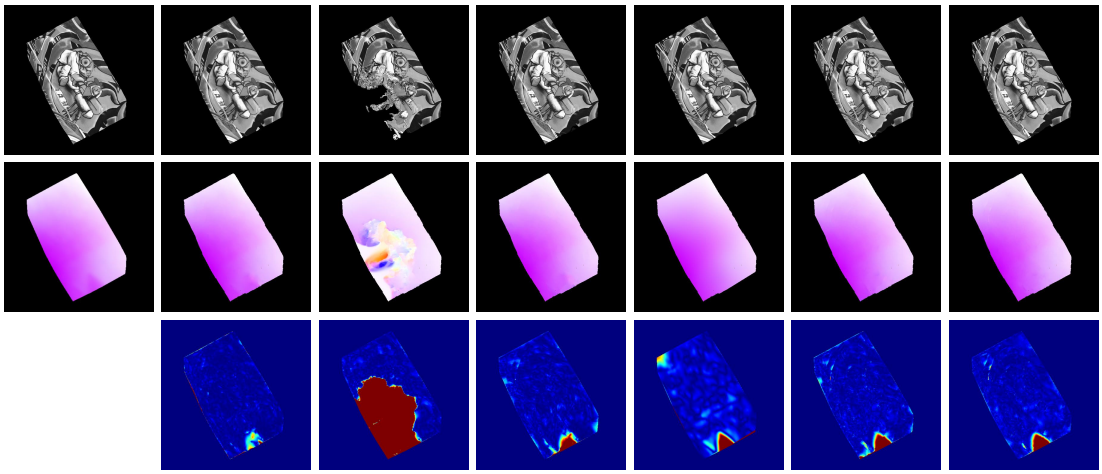
Methods	Computational Time (in Sec.)			
	Original	Occlusion	Guass.N	S&P.N
LME (Ours, Auto-meshing)	512.12	508.10	671.09	692.74

(d) Computational time on Garg *et al.* benchmark dataset [49].

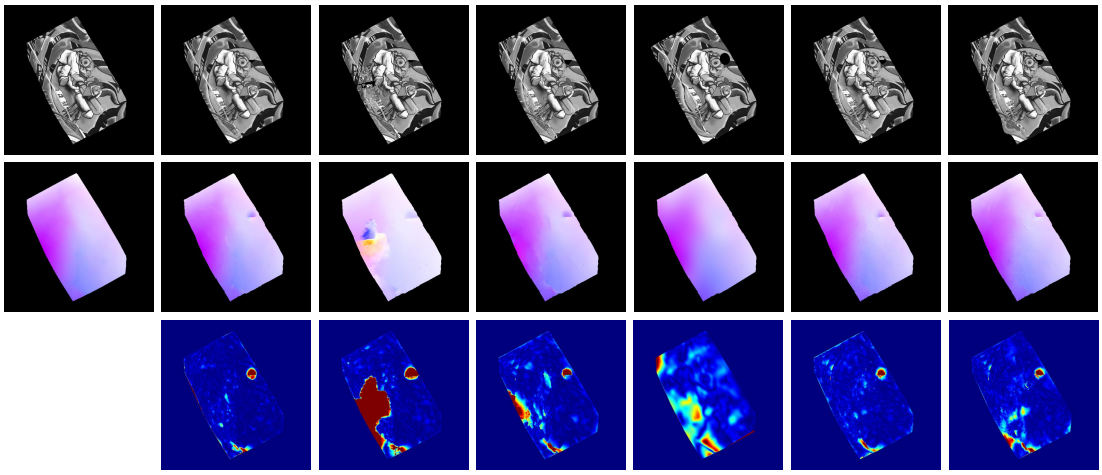


(e) Visual comparison on the alignment from the frame 30 to the reference frame in the sequence *S&P.Noise*.

Figure 3-7: Quantitative analysis (*Endpoint Error*) and the visual comparison on the Garg *et al.* benchmark dataset [49]. (a,b,c): *Average Endpoint Error* (AEE) and two robustness tests (R 1.0 and A75 [5]) are applied on results by varying methods. (d): The average computational time (in second) of our method. (e): **Top-left Boxes**: those include the chosen frame, the reference frame and their closed up. **The Rest**: the first row is the alignment results; the second row is the closeups; the third row is the error map against the ground truth flow field.



(a) The visual comparison on Frame 31 of sequence *Original* in Garg *et al.* benchmark dataset.

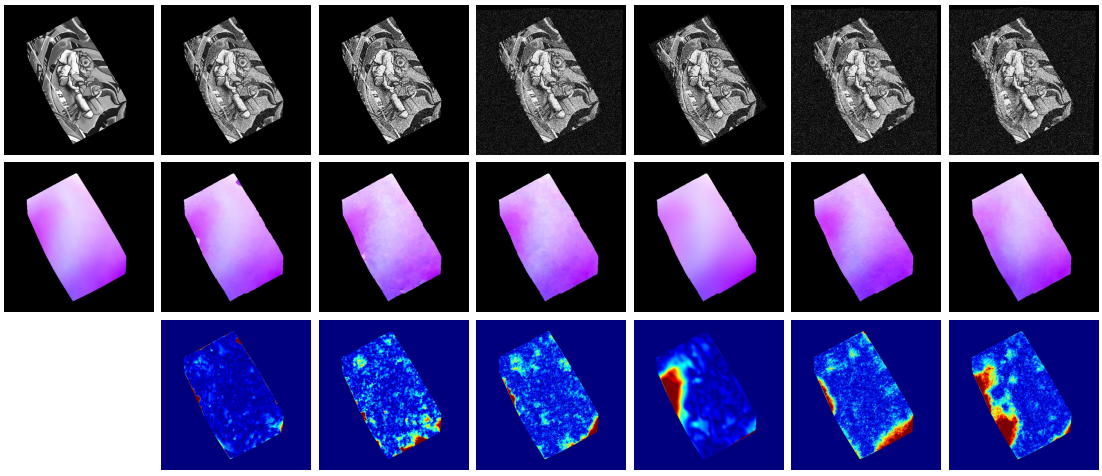


(b) The visual comparison on Frame 26 of sequence *Occlusion* in Garg *et al.* benchmark dataset.

Figure 3-8: Additional Visual Comparison on Sample Frames of *Original* and *Occlusion* in Garg *et al.* [49] Benchmark Dataset. (a): The reference frame and ground truth flow field. (b): *LME*. (c): Brox *et al.* [20] (d): ITV-L1 [143]. (e):Pizarro *et al.* [100]. (f): Garg *et al.*, DCT basis [49]. (g): Garg *et al.*, PCA basis [49]. **Rows from top to bottom**: The inverse warping result, the optical flow field and the error map.

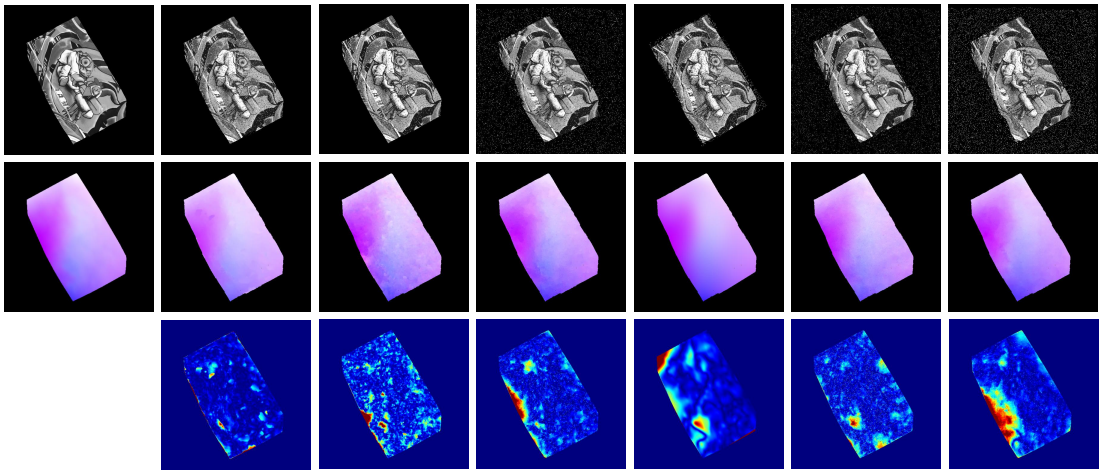
setting.

Fig. 3-7(a) shows *Average Endpoint Error* in pixel (AEE) on the four benchmark sequences of Garg *et al.*. *LME* displays the best AEE measurements on the *Original*, *Occlusion* and *S&P.Noise* sequences and outperforms Garg *et al.* (both PCA and DCT basis), ITV-L1 and Brox *et al.* algorithms on all four sequences. Pizarro *et al.* has comparable performance (slightly outperforming us by 0.25 RMS) to our method on the *Guass.Noise* sequence. In addition, we compute two robustness comparisons R 1.0 and A 75 in Fig. 3-7(b) and 3-7(c) respectively using identical approaches to those



(a) (b) (c) (d) (e) (f) (g)

(a) The visual comparison on Frame 24 of sequence *Gauss.Noise* in Garg *et al.* benchmark dataset.



(a) (b) (c) (d) (e) (f) (g)

(b) The visual comparison on Frame 25 of sequence *S&P.Noise* in Garg *et al.* benchmark dataset.

Figure 3-9: Additional Visual Comparison on Sample Frames of *Gauss.Noise* and *S&P.Noise* in Garg *et al.* [49] Benchmark Dataset. (a): The reference frame and ground truth flow field. (b): *LME*. (c): Brox *et al.* [20] (d): ITV-L1 [143]. (e):Pizarro *et al.* [100]. (f): Garg *et al.*, DCT basis [49]. (g): Garg *et al.*, PCA basis [49]. **Rows from top to bottom**: The inverse warping result, the optical flow field and the error map.

in [5]. *LME* yields the best performance in both R 1.0 and A 75 tests on all trials.

We also observe that the baselines of Garg *et al.*, Pizarro *et al.* and *LME* show competitive results to each other. The temporal subspace constraints of Garg *et al.* consider the 2D trajectories of different points across multiple images. Such trajectories correlation yields extra robustness to single object tracking but leads to potential error on the region where pixel trajectories may overlap on each other. Fig. 3-7(e) shows such a region (top left, blur square) where the pixels intuitively shrink together. Comparing to the remaining regions, Garg *et al.* provides large error (red in error map). On the

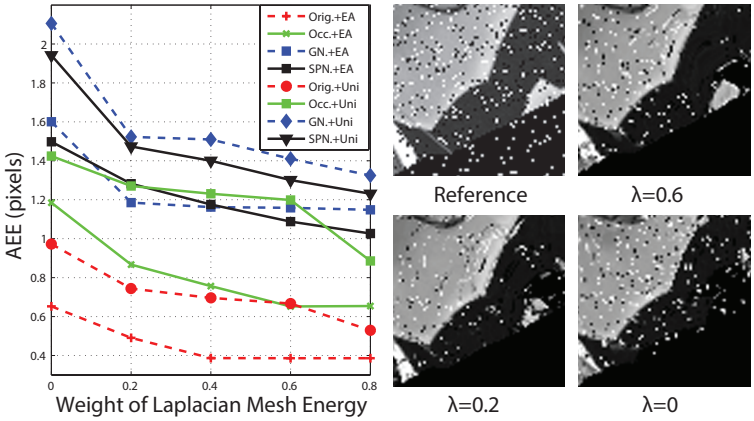


Figure 3-10: AEE measures on Garg *et al.* [49] benchmark sequences by varying the weighting λ (Edge-Aware (+EA) v.s. Uniform (+Uni) meshes). **Right**: Visual comparison of *LME* + *Edge-Aware* mesh on alignment from frame 30 to a reference in the sequence *S&P.Noise* by varying the weight λ.

other hand, *LME* takes into account Laplacian mesh deformation constraint which is encoded in a differential coordinate. Thus the change of actual pixel location yields less effect to the main energy. However, *LME* gives larger error in the noisy scenes. For instance, *LME* obtains the larger AEE (Fig. 3-7(a)) than Pizarro *et al.* and yields comparable performance over the other methods on the *Guass.Noise* sequence. We believe that this is because the large amount of Gaussian noise weakens control points detection in the *Detail-Aware Flow Field Enhancement* step (Sec. 3.3.2): the accuracy of SIFT feature detection and matching is thus reduced. This issue may cause inaccurate deformation of mesh $\mathcal{M}_2$ which could result in incorrect energy calculation within the function $E(w)$. One possible solution to this would be to use features more robust against noise or to use a low pass filter, which is left for future work.

Fig. 3-7(e) shows comparative *Inverse Image Warping* results between *LME* and five other state-of-the-art algorithms on all four Garg *et al.* benchmark sequences. Fig. 3-8 and 3-9 also give the quality comparison across the baselines, where the red colour presents larger error. Those examinations of the images illustrate that *Laplacian Mesh Energy* can generate a sharper and less distorted image after warping. This provides some insight into the algorithm's strong performance in the *Middlebury* interpolation result, as images warped using our computed flow appear to preserve local visual detail. Tab. 3-7(d) shows computational consumption of our method on Garg *et al.* benchmark sequences. Our method takes more computation on *Guass.Noise* and *S&P.Noise* because the image noises lead to more outliers which reduce the speed of mesh propagation (Sec. 3.3.2).

We also evaluate the effect of varying the weight of the *Laplacian Mesh Energy* on the Garg *et al.* dataset where λ is varied with discrete values between 0 and 1. As shown in Fig. 3-10, AEE is improved as the value λ increases on all trails. We observe that

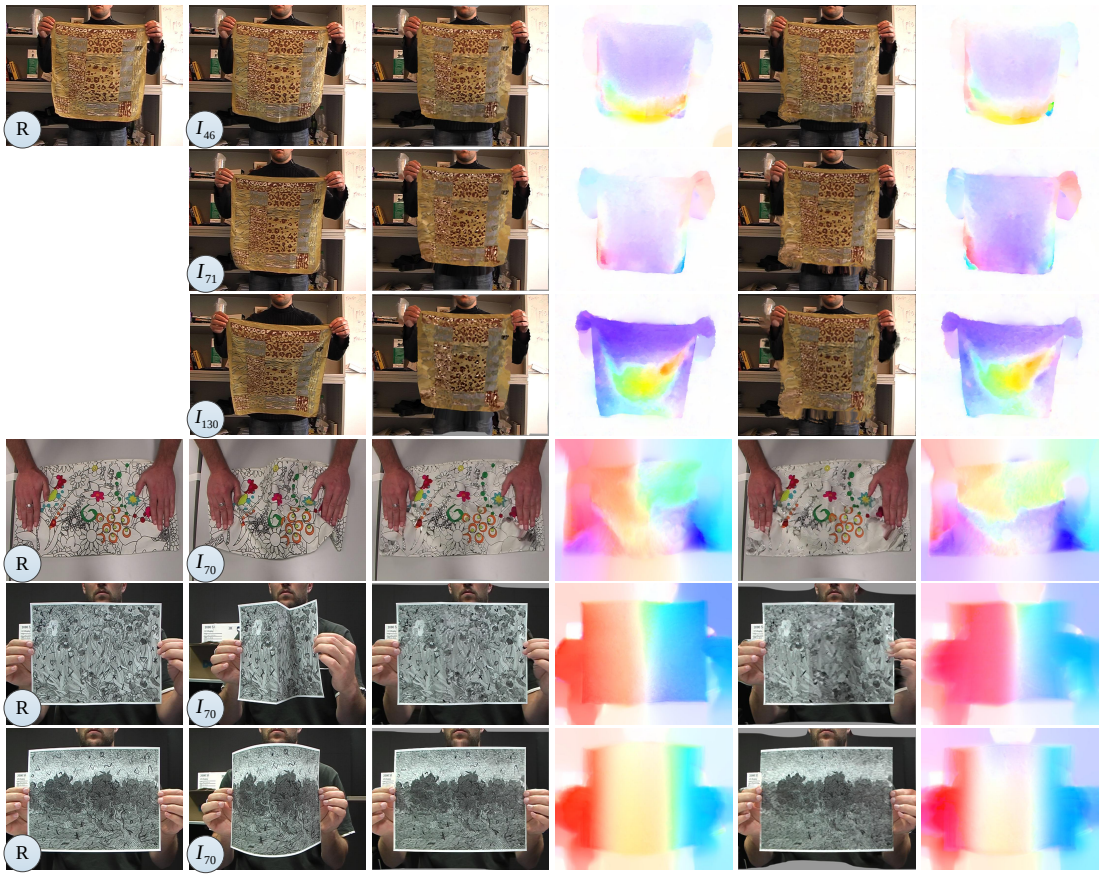


Figure 3-11: Visual Alignment Comparison on Real-World nonrigid Sequences *Cloth* [109], *Cushion*, *PaperCrease* and *PaperBend* [110]. (a): The reference frames. (b): The input frames. (c): The alignment result of *LME*. (d): The sum of concatenating flow fields computed by *LME*. (e): The alignment result of the baseline method. (f): The sum of concatenating flow fields computed by the baseline method.

even provided with a small weight (e.g. 0.2), *Laplacian Mesh Energy* still contributes a stronger preservation of the local flow structure and hence better preserved image detail during warping. When the increasing λ reaches 0.6 and even larger, we do not have significant improvement on most of the sequences. Thus we keep $\lambda = 0.6$ across all the experiments in the context. Furthermore, we also demonstrate how different input meshes may affect performance in Fig. 3-10 which shows a quantitative analysis of our method using *Edge-Aware* meshing (denoted by "+EA") and a uniform grid mesh (denoted by "+Uni", 5-pixel vertex distances) on the Garg *et al.* dataset. The former outperforms the uniform grid mesh in all four trails. It is because that – comparing with the unique grid mesh – the *Edge-Aware* mesh reduce the structure damage of grids on the object boundaries.

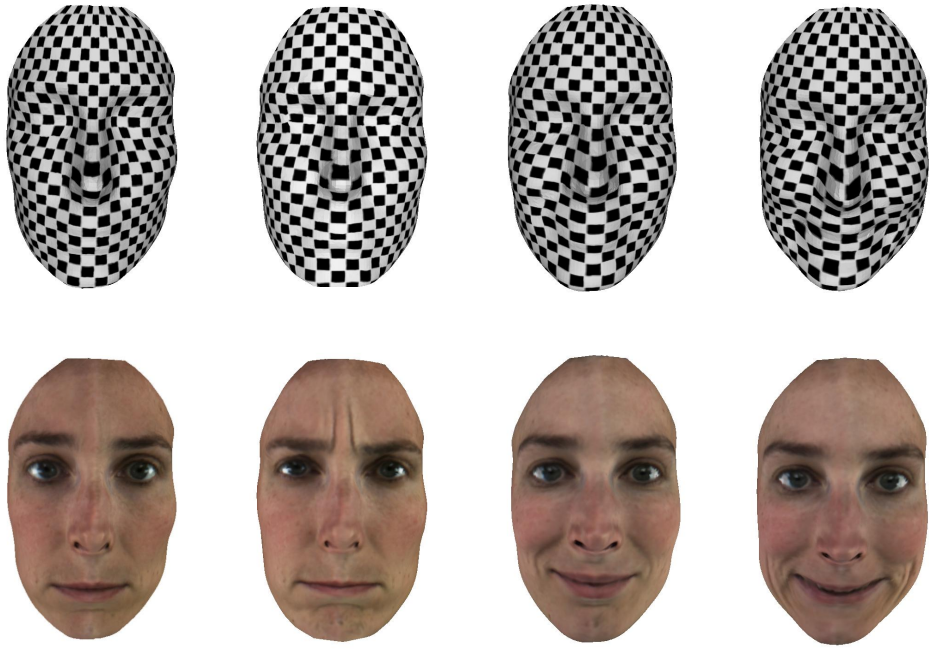


Figure 3-12: Example output from a *3D Dynamic Morphable Model*. **Top Row**: The checkered pattern highlights correct underlying mesh deformation, which is dependent on accurate nonrigid UV map registration. **Bottom Row**: Example images output from a *3D Dynamic Morphable Model*. **From Left To Right**: Sequences *AU-1+4+15*, *AU-4+7+17+23*, *AU-12+10* and *AU-20+23+25*.

3.4.3 Real-World Nonrigid Dataset

To validate the benefit of our *Laplacian Mesh Energy* on nonrigid sequences in particular the real-world cases, we consider a baseline method where we turn off the *Laplacian Mesh Energy* term ($\lambda = 0$) of *LME* and do not take any mesh as input. We compare this baseline method against our *LME* on several real-world sequences [109, 110] with nonrigid motion. For better observation on the effect of accumulated error for the small flow details, we introduce an alignment experiment as follows: First, each of the method is used to compute optical flow field for each adjacent pair of frames. The input frame is then warped back to the reference frame by concatenating flow fields. In this case, small errors in the optical flow field are accumulated then lead to more obvious artefacts into the final warpping results. As one can be observed in Fig. 3-11, the flow blur on the boundaries is significantly reduced by *LME* (*Edge-Aware* meshing strategy). Furthermore, the small image details inside the object are sharp and preserved during the alignment. The identical conclusion can also be observed on the three other real-world nonrigid sequences [109, 110].

3.4.4 3D Dynamic Morphable Model Construction

In this subsection, we show an application of our algorithm (*LME*) to the construction of *3D Dynamic Morphable Models* (3DDMM) [31]. These models are constructed from video-rate 3D facial scan data of different facial expressions. The essential problem with such data is aligning the 3D meshes such that all facial features are in correspondence. Solving this problem results in the same vertex topology deformed and tracked through the facial expression sequence. This can be approached by nonrigidly aligning the *UV Texture Maps* corresponding the face meshes to a reference texture (e.g. a neutral expression), and then generating the 3D correspondences from these aligned images [31]. We applied *LME* to the alignment of the *UV Texture Maps* for 6 dynamic facial sequences. After aligning the UV sequences, we constructed a *3D Dynamic Morphable Model* from the corresponding meshes and rendered the output sequences. Fig. 3-12 shows some example outputs, where a checkered pattern represents deformation in the underlying mesh. Note that more details can be found in the corresponding video footage of `http://www.cs.bath.ac.uk/~wl281/lme/LME-flow.mp4`.

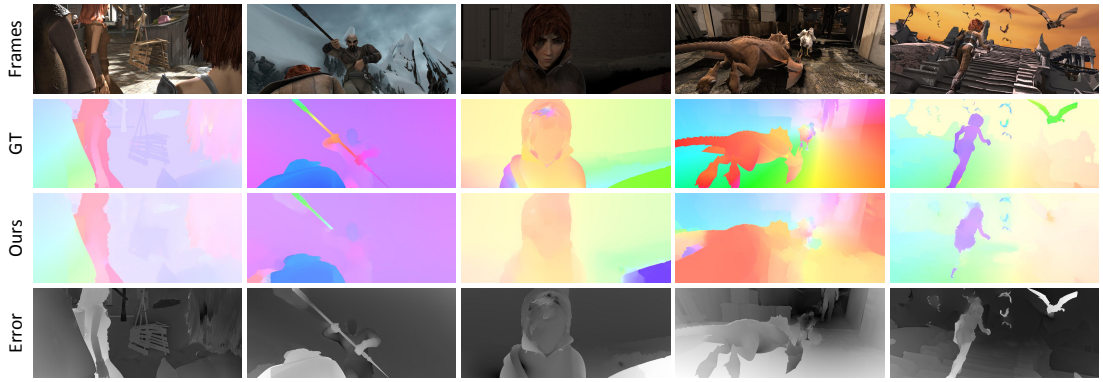
3.4.5 Sintel Dataset

In this subsection, we conduct the further investigation into the failure mode and suggest the application range of our method. As mentioned in the technical sections, our method takes into account both image properties and mesh geometry for the optical flow estimation. The implementation has shown the competitive accuracy in both Middlebury (Sec. 3.4.1) and Garg *et al.* Datasets (Sec. 3.4.2). The hidden conditions are that (1) the mesh edges have good fit onto the object boundaries; (2) the pixel displacement is small. In Fig. 3-5 shows the improvement using manually segmented mesh that gives better fit on the boundaries. Furthermore, our method cannot provide reasonable results if the pixel displacement between input images is larger than the distance between the neighbouring vertices. It is because that the large displacement may move vertices into other *neighbourhood ring*. Such broken structure may result in wrong energy into the main energy. Here we perform our method in the Sintel Dataset [23, 150], in order to investigate the failure for the large pixel displacement.

Sintel dataset contains two categories – *Clean* and *Final* – each of which includes 12 long synthetic sequences. The *Clean* pass mainly contains the various properties of changing illumination, shadow and smooth surface shading, while the *Final* pass contains all sequences from the *Clean* pass but added more difficult atmospheric effects of depth of field blur and motion blur, etc. In general, Sintel dataset is more difficult than Middlebury and Garg *et al.* because it contains very large pixel displacement (larger than 40 pixel) and geometric blur. Both of these issues are still unsolved in the optical flow community. Fig. 3-13(a) shows the evaluation of our method (LME) in Sintel

Methods	Evaluation on Sintel Dataset						
	Time (in second)	EPE	matched	unmatched	s0-10	s10-40	s40+
MDP [152]	N/A	8.445	4.150	43.430	1.420	5.449	50.507
LDOF [21]	N/A	9.116	5.037	42.344	1.485	4.839	57.296
LME (Ours, Auto-meshing)	1261.32	13.064	8.897	46.933	2.442	12.412	66.991

(a) Evaluation of different methods on Sintel Dataset [23] by 11th Feb. 2014.



(b) Visual comparison of our method on sample frames from *Clean* pass of Sintel dataset.

Figure 3-13: Quantitative *Endpoint Error* (EPE) analysis and the visual comparison on Sintel dataset [49].

dataset where several metrics[2] – EPE, matched, unmatched, s0-10, s10-40 and s40+ – are performed. EPE denotes the overall *Endpoint Error*; matched is the *Endpoint Error* on the unoccluded region while unmatched presents the *Endpoint Error* on the occluded one; s0-10 denotes the *Endpoint Error* on the region with pixel displacement smaller than 10 pixels; s10-40 is the *Endpoint Error* on the region with pixel displacement between 10 and 40 pixels; s40+ denotes the *Endpoint Error* on the region with pixel displacement larger than 40 pixels. Note that we shows only MDP and LDOF as baseline methods because the results of Garg *et al.*, Pizarro *et al.*, ITV-L1 and Brox *et al.* are not available in the Sintel evaluation website. Our method shows large errors on all the trials including the s0-10 and s40+. It is because that most frames of S-intel contain both small and large pixel displacement. The latter destroys the mesh structure and yields large wrong energy. Such situation harms the energy optimisation on all the optical flow (including small displacement ones and large ones) within those broken *neighbour rings*. The possible improvement could be using sparse mesh where the average distance between neighbour vertices is larger than the maximum pixel displacement. However, the sparse mesh may weaken the constraint on the small motion regions. This unsolved issue is left as the potential future research.

[2] http://sintel.is.tue.mpg.de/results/

3.5 Conclusion

In this chapter we have presented a novel optical flow approach which uses *Laplacian Mesh Energy* to preserve local continuity of optical flow estimated on nonrigid deformations. Adapted from computer graphics, our novel energy achieves this property by minimising differentials in Laplacian coordinates. In our evaluation we have compared our method to several state-of-the-art optical flow approaches on two well known evaluation sets. It has been demonstrated that our algorithm is capable of providing accurate flow estimation and also preserving local image detail – evident through high scores in Middlebury evaluation, and comparison to Garg *et al.*. For future work we are interested in more intelligently creating the underlying mesh to better approximate the image and motion of interest.

The related publication is shown as follows:

[77] **W. Li**, D. Cosker, M. Brown, and R. Tang, *Optical Flow Estimation using Laplacian Mesh Energy*, in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), IEEE, June 2013, pp. 2435–2442.

Chapter 4

Robust Dense Tracking in Blurred Scenes

Optical flow estimation is a difficult task given real-world video footage with camera shake and object motion blur. In this chapter, we combine a commercial 3D pose&position tracker with an RGB sensor allowing us to capture video footage together with 3D camera motion. We show that the additional tracked camera trajectory can be embedded into a hybrid optical flow framework by interleaving an iterative blind deconvolution and warping based minimisation scheme. Such a hybrid framework significantly improves the accuracy of optical flow estimation in scenes with strong blur. Our approach yields improved overall performance against three state-of-the-art baseline methods applied to our proposed ground truth sequences as well as in several other real-world sequences captured by our novel imaging system.

4.1 Introduction

Camera shake blur often occurs during fast camera movement in low-light conditions due to the requirement of adopting a longer exposure. Recovering both the blur kernel and the latent image from a single blurred image is known as *Blind Deconvolution* which is an inherently ill-posed problem. Cho and Lee [28] propose a fast deblurring process within a coarse-to-fine framework (Cho&Lee) using a predicted edge map as a prior. To reduce the noise effect in this framework, Zhong *et al.* [158] introduce a pre-filtering process which reduces the noise along a specific direction and preserves the image information in other directions. Their improved framework provides high quality kernel estimation with a low run-time but shows difficulties given combined object and camera shake blur.

To obtain higher performance, a handful of combined hardware and software-based approaches have also been proposed for image deblurring. Tai *et al.* [133] introduce a hybrid imaging system that is able to capture both video at high frame rate and a blurry image. The optical flow fields between the video frames are utilised to guide

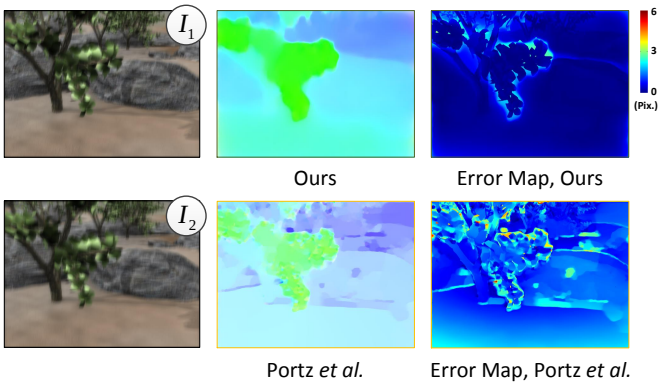


Figure 4-1: Visual comparison of our method to Portz *et al.* [101] on our ground truth benchmark *Grove2* with synthetic camera shake blur. **First Column**: the input images; **Second Column**: the optical flow fields calculated by our method and the baseline; **Third Column**: the RMS error maps against the ground truth.

blur kernel estimation. Levin *et al.* [70] propose to capture a uniformly blurred image by controlling the camera motion along a parabolic arc. Such uniform blur can then be removed based on the speed or direction of the known arc motion. As a complement to Levin *el al.*'s [70] hardware-based deblurring algorithm, Joshi *et al.* [60] apply inertial sensors to capture the acceleration and angular velocity of a camera over the course of a single exposure. This extra information is introduced as a constraint in their energy optimisation scheme for recovering the blur kernel. All the hardware-assisted solutions described provide extra information in addition to the blurry image, which significantly improves overall performance. However, the methods require complex electronic setups and the precise calibration.

Optical flow techniques are widely studied and adopted across computer vision. One of advantages is the dense image correspondences they provide. In the last two decades, the optical flow model has evolved extensively – one landmark work being the variational model of Horn and Schunck [55] where the concept of *Brightness Constancy* is proposed. Under this assumption, pixel intensity does not change spatio-temporally, which is, however, often weakened in real-world images because of natural noise. To address this issue, some complementary concepts have been developed to improve performance given large displacements [20], taking advantage of feature-rich surfaces [153] and adapting to nonrigid deformation in scenes (Chapter 3). However, flow approaches that can perform well given blurred scenes – where the *Brightness Constancy* is usually violated – are less common. Of the approaches that do exist, Schoueri *et al.* [114] perform a linear deblurring filter before optical flow estimation while Portz *et al.* [101] attempt to match un-uniform camera motion between neighbouring input images. Whereas the former approach may be limited given nonlinear blur in real-world scenes; the latter requires two extra frames to parameterise the motion-induced blur. Regarding non optical-flow based methods, Yuan *et al.* [156] align a blurred image to

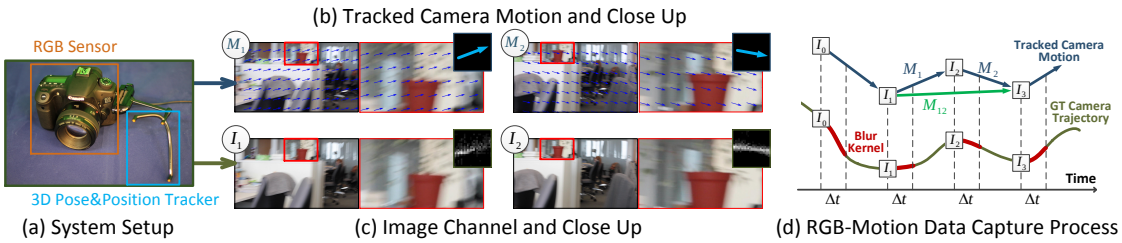


Figure 4-2: RGB-Motion Imaging System. **(a)**: Our system setup using a combined RGB sensor and 3D Pose&Position Tracker. **(b)**: The tracked 3D camera motion in relative frames. The top-right box is the average motion vector – which has similar direction to the blur kernel. **(c)**: Images captured from our system. The top-right box presents the blur kernel estimated using [28]. **(d)**: The internal process of our system where the Δt presents the exposure time.

a sharp one by predefining an affine image transform with a blur kernel. Similarly HaCohen *et al.* [52] achieve alignment between a blurred image and a sharp one by embedding deblurring into the correspondence estimation.

4.1.1 Contributions

Our major contribution in this chapter is to utilise an *RGB-Motion Imaging System* – an RGB sensor combined with a 3D pose&position tracker – in order to propose: **(A)** an iterative enhancement process for camera shake blur estimation which encompasses the tracked camera motion (Sec. 4.2) and a *Directional High-pass Filter* (Sec. 4.3 and Sec. 4.6.2); **(B)** a *Blur-Robust Optical Flow Energy* formulation (Sec. 4.5); and **(C)** a hybrid coarse-to-fine framework (Sec. 4.6) for computing optical flow in blur scenes by interleaving an iterative blind deconvolution process and a warping based minimisation scheme. In the evaluation section, we compare our method to three existing state-of-the-art optical flow approaches on our proposed ground truth sequences (Fig. 4-1, containing blur and baseline blur-free equivalents) and also illustrate the practical benefit of our algorithm given other real-world cases.

4.2 RGB-Motion Imaging System

Camera shake blur within video footage is typically due to fast camera motion and/or long exposure time. In particular, such blur can be considered as a function of the camera trajectory supplied to image space during the exposure time Δt. It therefore follows that knowledge of the actual camera motion between image pairs can provide significant information when performing image deblurring [60, 70].

In this chapter, we propose a simple and portable setup (Fig. 4-2(a)), combining an RGB sensor and a 3D pose&position tracker (SmartNav by NaturalPoint Inc.) in order to capture continuous scenes (video footage) along with real-time camera pose&position information. Note that the RGB sensor could be any camera or a Kinect sensor – A

Canon EOS 60D is applied in our implementation to capture 1920×1080 video at frame rate of 24 FPS. Furthermore, our tracker is proposed to provide the rotation (yaw, pitch and roll), translation and zoom information within a reasonable error range (2 mm). To synchronise this tracker data and the image recording, a real time collaboration (RTC) server [68] is built using the instant messaging protocol XMPP (also known as Jabber[1]) which is designed for message-oriented communication based on XML, and allows real-time responses between different messaging channels or any signal channels that can be transmitted and received in message form. In this case, a time stamp is assigned to the received message package by the central timer of the server. Those message packages are synchronised if they contain nearly the same time stamp. We consider the Jabber for synchronisation because of its opensource nature and the low respond delay (around 10 ms).

Assuming objects have similar depth within the same scene (a common assumption in image deblurring which will be discussed in our future work), the tracked 3D camera motion in image coordinates can be formulated as:

$$\mathbf{M}_j = \frac{1}{n} \sum_{\mathbf{x}} K \left([R|T] \, \mathbf{X}_{j+1} - \mathbf{X}_j \right) \tag{4.1}$$

where $\mathbf{M}_j$ represents the average of the camera motion vectors from the image j to image $j+1$. $\mathbf{X}$ denotes the 3D position of the camera while $\mathbf{x} = (x, y)^T$ is a pixel location and n represents the number of pixels in an image. K represents the 3D projection matrix while R and T denote the rotation and translation matrices respectively of tracked camera motion in the image domain. All these information K, R and T is computed using `Optitrack`'s *Camera SDK*[2] (version 1.2.1). Fig 4-2(b,c) shows sample data (video frames and camera motion) captured from our imaging system. It is observed that blur from the real-world video is near linear due to the relatively high sampling rate of the camera. The blur direction can therefore be approximately described using the tracked camera motion. Let the tracked camera motion $\mathbf{M}_j = (r_j, \theta_j)^T$ be represented in polar coordinates where r_j and θ_j denote the magnitude and directional component respectively. j is a sharing index between tracked camera motion and frame number. In addition, we also consider the combined camera motion vector of neighbouring images as shown in Fig 4-2(d), e.g. $\mathbf{M}_{12} = \mathbf{M}_1 + \mathbf{M}_2$ where $\mathbf{M}_{12} = (r_{12}, \theta_{12})$ denotes the combined camera motion vector from image 1 to image 3. As one of our main contributions, these real-time motion vectors are proposed to provide additional constraints for blur kernel enhancement (Sec. 4.6) within our optical flow framework.

[1] http://www.jabber.org/
[2] http://www.naturalpoint.com/optitrack

4.3 Blind Deconvolution

The motion blur process can commonly be formulated:

$$I = k \otimes l + n \tag{4.2}$$

where I is a blurred image and k represents a blur kernel w.r.t. a specific *Point Spread Function*. l is the latent image of I; $\otimes$ denotes the convolution operation and n represents spatial noise within the scene. In the blind deconvolution operation, both k and l are estimated from I, which is an ill-posed (but extensively studied) problem. A common approach for blind deconvolution is to solve both k and l in an iterative framework using a coarse-to-fine strategy:

$$k = \mathbf{argmin}_k\{\|I - k \otimes l\| + \rho(k)\}, \tag{4.3}$$

$$l = \mathbf{argmin}_l\{\|I - k \otimes l\| + \rho(l)\}. \tag{4.4}$$

where ρ represents a regularization that penalizes spatial smoothness with a sparsity prior [28], and is widely used in recent state-of-the-art work [118, 153]. Due to noise sensitivity, low-pass and bilateral filters [134] are typically employed before deconvolution. Eq. 4.5 denotes the common definition of an optimal kernel from a filtered image.

$$\begin{aligned} k_f &= \mathbf{argmin}_{k_f}\{\|(k \otimes l + n) \otimes f - k_f \otimes l\| + \rho(k_f)\} \\ &\approx \mathbf{argmin}_{k_f} \|l \otimes (k \otimes f - k_f)\| = k \otimes f \end{aligned} \tag{4.5}$$

where k represents the ground truth blur kernel, f is a filter, and k_f denotes the optimal blur kernel from the filtered image $I \otimes f$. The low-pass filtering process improves deconvolution computation by removing spatially-varying high frequency noise but also results in the removal of useful information which yields additional errors over object boundaries. To preserve this useful information, we introduce a directional high-pass filter that utilises our tracked 3D camera motion.

4.4 Directional High-pass Filter

Detail enhancement using directional filters has been proved effective in several areas of computer vision [158]. Here we define a directional high-pass filter as:

$$f_\theta \otimes I(\mathbf{x}) = m \int g(t) I(\mathbf{x} + t\Theta) dt \tag{4.6}$$

where $\mathbf{x} = (x, y)^T$ represents a pixel position and $g(t) = 1 - exp\{-t^2/2\sigma^2\}$ denotes a 1D Gaussian based high-pass function. $\Theta = (\cos\theta, \sin\theta)^T$ controls the filtering

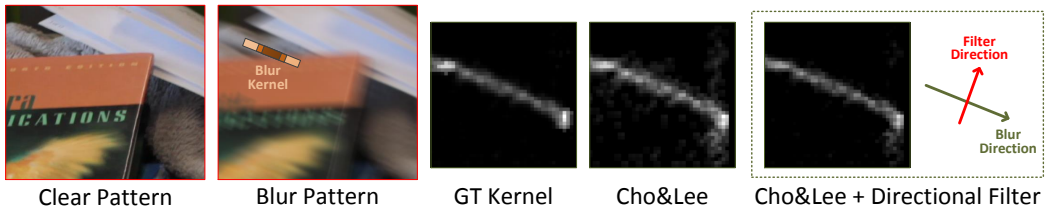


Figure 4-3: Directional high-pass filter for blur kernel enhancement. Given the blur direction θ, a directional high-pass filter along $\theta + \pi/2$ is applied to preserve blur detail in the estimated blur kernel.

direction along θ. m is a normalization factor defined as $m = \left(\int g(t)dt\right)^{-1}$. The filter f_θ is proposed to preserve overall high frequency details along direction θ without affecting blur detail in orthogonal directions [26]. Given a directionally filtered image $b_\theta = f_\theta \otimes I(\mathbf{x})$, the optimal blur kernel is defined (Eq 4.5) as $k_\theta = k \otimes f_\theta$. Fig. 4-3 demonstrates that noise or object motion within a scene usually results in low frequency noise in the estimated blur kernel (Cho&Lee [28]). This low frequency noise can be removed by our directional high-pass filter while preserving major blur details. In our method, this directional high-pass filter is supplemented into the Cho&Lee [28] framework using a coarse-to-fine strategy in order to recover high quality blur kernels for use in our optical flow estimation (Sec. 4.6.2).

4.5 Blur-Robust Optical Flow Energy

Within a blurry scene, a pair of adjacent natural images may contain different blur kernels, further violating *Brightness Constancy*. This results in unpredictable flow error across the different blur regions. To address this issue, Portz *et al.* proposed a modified *Brightness Constancy* term by matching the un-uniform blur between the input images. As one of our main contributions, we extend this assumption to a novel *Blur Gradient Constancy* term in order to provide extra robustness against illumination change and outliers. Our main energy function is given as follows:

$$E(\mathbf{w}) = E_B(\mathbf{w}) + \gamma E_S(\mathbf{w}) \tag{4.7}$$

A pair of consecutively observed frames from an image sequence is considered in our algorithm. $I_1(\mathbf{x})$ represents the current frame and its successor is denoted by $I_2(\mathbf{x})$ where $I_* = k_* \otimes l_*$ and $\{I_*, l_* : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}\}$ represent rectangular images in the RGB channel. Here l_* is latent image and k_* denotes the relative blur kernel. The optical flow displacement between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ is defined as $\mathbf{w} = (u, v)^T$. To match the un-uniform blur between input images, the blur kernel from each input image is applied to the other. We have new blur images b_1 and b_2 as follows:

$$b_1 = k_2 \otimes I_1 \approx k_2 \otimes k_1 \otimes l_1 \tag{4.8}$$

$$b_2 = k_1 \otimes I_2 \approx k_1 \otimes k_2 \otimes l_2 \tag{4.9}$$

Our energy term encompassing *Brightness* and *Gradient Constancy* relates to b_1 and b_2 as follows:

$$E_B(\mathbf{w}) = \int_\Omega \phi(\|b_2(\mathbf{x}+\mathbf{w}) - b_1(\mathbf{x})\|^2 + \alpha \|\nabla b_2(\mathbf{x}+\mathbf{w}) - \nabla b_1(\mathbf{x})\|^2) d\mathbf{x} \tag{4.10}$$

The term $\nabla = (\partial_{xx}, \partial_{yy})^T$ presents a spatial gradient and $\alpha \in [0, 1]$ denotes a linear weight. The smoothness regulariser penalizes global variation as follows:

$$E_S(\mathbf{w}) = \int_\Omega \phi(\|\nabla u\|^2 + \|\nabla v\|^2) d\mathbf{x} \tag{4.11}$$

where we apply the Lorentzian regularisation $\phi(s) = log(1+s^2/2\epsilon^2)$ to both the data term and smoothness term. In our case, the image properties, e.g. small details and edges, are broken by the camera blur, which leads to additional errors in those regions. We suppose to apply strong boundary preservation even the non-convex Lorentzian regularisation may bring the extra difficulty to the energy optimisation (More analysis can be found in Sec.2.1.3 and Tab. 2-1). In the following section, our optical flow framework is introduced in detail.

4.6 Optical Flow Framework

Our overall framework is outlined in Algorithm 1 based on an iterative top-down, coarse-to-fine strategy. Prior to minimizing the *Blur-Robust Optical Flow Energy* (Sec. 4.6.4), a fast blind deconvolution approach [28] is performed for pre-estimation of the blur kernel (Sec. 4.6.1), which is followed by kernel refinement using our *Directional High-pass Filter* (Sec. 4.6.2). All these steps are detailed in the following subsections.

4.6.1 Iterative Blind Deconvolution

Cho and Lee [28] describe a fast and accurate approach (Cho&Lee) to recover the unique blur kernel. As shown in Algorithm 1, we perform a similar approach for the pre-estimation of the blur kernel k within our iterative process, which involves two steps of prediction and kernel estimation. Given the latent image l estimated from the consecutively coarser level, the gradient maps $\Delta l = \{\partial_x l, \partial_y l\}$ of l are calculated along the horizontal and vertical directions respectively in order to enhance salient edges and reduce noise in featureless regions of l. Next, the predicted gradient maps Δl as well

Algorithm 1: Blur-Robust Optical Flow Framework

Input : A image pair I_1, I_2 and camera motion θ_1, θ_2, θ_{12}
Output : Optimal optical flow field $\mathbf{w}$

A n-level top-down pyramid is built with the level index i
$i \leftarrow 0$
$l_1^i \leftarrow I_1^i$, $l_2^i \leftarrow I_2^i$
$k_1^i \leftarrow 0$, $k_2^i \leftarrow 0$, $\mathbf{w}^i \leftarrow (0,0)^T$
for *coarse to fine* **do**
    $i \leftarrow i + 1$
    Resize $k_{\{1,2\}}^i$, $l_{\{1,2\}}^i$, $I_{\{1,2\}}^i$ and $\mathbf{w}^i$ with the ith scale
    foreach $* \in \{1, 2\}$ **do**
        $k_*^i \leftarrow$ `IterativeBlindDeconvolve` (l_*^i, I_*^i)
        $k_*^i \leftarrow$ `DirectionalFilter` ($k_*^i, \theta_1, \theta_2, \theta_{12}$)
        $l_*^i \leftarrow$ `NonBlindDeconvolve` (k_*^i, I_*^i)
    endfor
    $b_1^i \leftarrow I_1^i \otimes k_2^i$, $b_2^i \leftarrow I_2^i \otimes k_1^i$
    $d\mathbf{w}^i \leftarrow$ `Energyoptimisation` ($b_1^i, b_2^i, \mathbf{w}^i$)
    $\mathbf{w}^i \leftarrow \mathbf{w}^i + d\mathbf{w}^i$
endfor

as the gradient map of the blurry image I are utilised to compute the pre-estimated blur kernel by minimizing the energy function as follows:

$$
\begin{aligned}
k = \mathbf{argmin}_k \sum_{I_*, l_*} \omega_* \left\| I_* - k \otimes l_* \right\|^2 + \delta \left\| k \right\|^2 \\
(I_*, l_*) \in \{(\partial_x I, \partial_x l), (\partial_y I, \partial_y l), (\partial_{xx} I, \partial_{xx} l), \\
(\partial_{yy} I, \partial_{yy} l), (\partial_{xy} I, (\partial_x \partial_y + \partial_y \partial_x) l/2)\}
\end{aligned} \tag{4.12}
$$

where δ denotes the weight of Tikhonov regularization and $\omega_* \in \{\omega_1, \omega_2\}$ represents a linear weight for the derivatives in different directions. Both I and l are propagated from the nearest coarse level within the pyramid. To minimise this energy Eq. (4.12), we follow the inner-iterative numerical scheme of [28] which yields a pre-estimated blur kernel k.

4.6.2 Directional High-pass Filtering

Once the pre-estimated kernel k is obtained, our *Directional High-pass Filters* are applied to enhance the blur information by reducing noise in the orthogonal direction of the tracked camera motion. Although our *RGB-Motion Imaging System* provides an intuitively accurate camera motion estimation, outliers may still exist in the synchronisation. We take into account the directional components $\{\theta_1, \theta_2, \theta_{12}\}$ of two consecutive camera motions M_1 and M_2 as well as their combination M_{12} (Fig. 4-2(d)) for extra

robustness. The pre-estimated blur kernel is filtered along its orthogonal direction as follows:

$$k = \sum_{\beta_*, \theta_*} \beta_* k \otimes f_{\theta_* + \pi/2} \tag{4.13}$$

where $\beta_* \in \{1/2, 1/3, 1/6\}$ linearly weights the contribution of filtering in different directions. Note that two consecutive images I_1 and I_2 are involved in our framework where the former accepts the weight set $(\beta_*, \theta_*) \in \{(1/2, \theta_1), (1/3, \theta_2), (1/6, \theta_{12})\}$ while the other weight set $(\beta_*, \theta_*) \in \{(1/3, \theta_1), (1/2, \theta_2), (1/6, \theta_{12})\}$ is performed for the latter. This filtering process yields an updated blur kernel k which is used to update the latent image l within a non-blind deconvolution [158]. Note that the convolution operation is computationally expensive in the spatial domain, we consider an equivalent filtering scheme in the frequency domain in the following subsection.

4.6.3 Convolution for Directional Filtering

Our proposed directional filtering is performed as convolution operation in the spatial domain, which is often highly expensive in computation given large image resolutions. In our implementations, we consider a directional filtering scheme in the frequency domain where we have the equivalent form of filtering model Eq. (4.6) as follows:

$$K_\Theta(u, v) = K(u, v) F_\Theta(u, v) \tag{4.14}$$

where K_Θ is the optimal blur kernel in the frequency domain while K and F_Θ present the *Fourier Transform* of the blur kernel k and our directional filter f_θ respectively. Thus, the optimal blur kernel k_θ in the spatial domain can be calculated as $k_\theta = \texttt{IDFT}[K_\Theta]$ using *Inverse Fourier Transform*. In this case, the equivalent form of our directional high-pass filter in the frequency domain is defined as follows:

$$F_\Theta(u, v) = 1 - exp\left\{-L^2(u, v)/2\sigma^2\right\} \tag{4.15}$$

where the line function $L(u, v) = u \cos\theta + v \sin\theta$ controls the filtering process along the direction θ while σ is the standard deviation for controlling the strength of the filter. Please note that other more sophisticated high-pass filters could also be employed using this directional substitution L. Even though this consumes a reasonable proportion of computer memory, convolution in the frequency domain $O(N \log_2 N)$ is faster than equivalent computation in the spatial domain $O(N^2)$.

Having performed blind deconvolution and directional filtering (Sec. 4.6.1, 4.6.2 and 4.6.3), two updated blur kernels k_1^i and k_2^i on the ith level of the pyramid are obtained from input images I_1^i and I_2^i respectively, which is followed by the uniform blur image

b_1^i and b_2^i computation using Eq. (4.9). In the following subsection, *Blur-Robust Optical Flow Energy* optimisation on b_1^i and b_1^i is introduced in detail.

4.6.4 Optical Flow Energy optimisation

As mentioned in Sec. 4.5, our blur-robust energy is continuous but highly nonlinear. minimisation of such energy function is extensively studied in the optical flow community. In this section, a numerical scheme combining *Euler-Lagrange Equations* and *Nested Fixed Point Iterations* is applied [20] to solve our main energy function Eq. 4.7. For clarity of presentation, we define the following mathematical abbreviations:

$$
\begin{aligned}
b_x &= \partial_x b_2(\mathbf{x}+\mathbf{w}) & b_{yy} &= \partial_{yy} b_2(\mathbf{x}+\mathbf{w}) \\
b_y &= \partial_y b_2(\mathbf{x}+\mathbf{w}) & b_z &= b_2(\mathbf{x}+\mathbf{w}) - b_1(\mathbf{x}) \\
b_{xx} &= \partial_{xx} b_2(\mathbf{x}+\mathbf{w}) & b_{xz} &= \partial_x b_2(\mathbf{x}+\mathbf{w}) - \partial_x b_1(\mathbf{x}) \\
b_{xy} &= \partial_{xy} b_2(\mathbf{x}+\mathbf{w}) & b_{yz} &= \partial_y b_2(\mathbf{x}+\mathbf{w}) - \partial_y b_1(\mathbf{x})
\end{aligned}
$$

After *Euler-Lagrange Equations* are applied to Eq. (4.7), we minimise the resulting system in a coarse-to-fine framework within a top-down image pyramid. In the outer fixed point iterations, we initialize the flow field $\mathbf{w} = (0,0)^T$ on the top (coarsest) level of the pyramid and propagate this to the next finer level as $\mathbf{w}^{i+1} \approx \mathbf{w}^i + d\mathbf{w}^i$ where we follow the assumption that the flow field on finer level $i+1$ is estimated by the flow field and the increments from the previous coarser level k. First order *Taylor Expansion* is then applied to the terms of b_z^{i+1}, b_{xz}^{i+1} and b_{yz}^{i+1}, which results in

$$
\begin{aligned}
b_z^{i+1} &\approx b_z^i + b_x^i du^i + b_y^i dv^i, \\
b_{xz}^{i+1} &\approx b_{xz}^k + b_{xx}^i du^i + b_{xy}^i dv^i, \\
b_{yz}^{i+1} &\approx b_{yz}^k + b_{xy}^i du^i + b_{yy}^i dv^i.
\end{aligned}
$$

where du^i and dv^i are two unknown increments which will be solved in our inner fixed point iterations. Given the initialization of $du^{i,0} = 0$ and $dv^{i,0} = 0$, we assume that $du^{i,j}$ and $dv^{i,j}$ converge within j iterations. We have the final linear system in $du^{i,j+1}$ and $dv^{i,j+1}$ as follows:

$$
\begin{aligned}
&(\phi')_B^{i,j} \cdot \{b_x^i(b_z^i + b_x^i du^{i,j+1} + b_y^i dv^{i,j+1}) \\
&+\alpha b_{xx}^i(b_{xz}^i + b_{xx}^i du^{i,j+1} + b_{xy}^i dv^{i,j+1}) \\
&+\alpha b_{xy}^i(b_{yz}^i + b_{xy}^i du^{i,j+1} + b_{yy}^i dv^{i,j+1})\} \\
&\qquad -\gamma\ (\phi')_S^{i,j} \cdot \nabla(u^i + du^{i,j+1}) = 0
\end{aligned} \tag{4.16}
$$

$$
\begin{aligned}
&(\phi')_B^{i,j} \cdot \{b_y^i(b_z^i + b_x^i du^{i,j+1} + b_y^i dv^{i,j+1}) \\
&+\alpha b_{yy}^i(b_{yz}^i + b_{xy}^i du^{i,j+1} + b_{yy}^i dv^{i,j+1}) \\
&+\alpha b_{xy}^i(b_{xz}^i + b_{xx}^i du^{i,j+1} + b_{xy}^i dv^{i,j+1})\} \\
&\qquad -\gamma\ (\phi')_S^{i,j} \cdot \nabla(v^i + dv^{i,j+1}) = 0
\end{aligned} \tag{4.17}
$$

where $(\phi')_B^{i,j}$ denotes a robustness factor against flow discontinuity and occlusion on the object boundaries. $(\phi')_S^{i,j}$ represents the diffusivity of the smoothness regularization.

$$
\begin{aligned}
(\phi')_B^{i,j} &= \phi'\{(b_z^i + b_x^i du^{i,j} + b_y^{i,j} dv^{i,j})^2 \\
&\quad + \alpha(b_{xz}^i + b_{xx}^i du^{i,j} + b_{xy}^i dv^{i,j})^2 \\
&\quad + \alpha(b_{yz}^i + b_{xy}^i du^{i,j} + b_{yy}^i dv^{i,j})^2\} \\
(\phi')_S^{i,j} &= \phi'\{\left\|\nabla(u^i + du^{i,j})\right\|^2 + \left\|\nabla(v^i + dv^{i,j})\right\|^2\}
\end{aligned}
$$

In our implementation, the image pyramid is constructed with a downsampling factor of 0.75. The final linear system in Eq. (4.16,4.17) is solved using *Conjugate Gradients* within 45 iterations.

4.7 Evaluation

In this section, we evaluate our method on both synthetic and real-world sequences and compare its performance against three existing state-of-the-art optical flow approaches of Xu *et al.*'s MDP [153], Portz *et al.*'s [101] and Brox *et al.*'s [20] (an implementation of [80]). MDP is one of the best performing optical flow methods given blur-free scenes, and is one of the top 3 approaches in the Middlebury benchmark [5]. Portz *et al.*'s method represents the current state-of-the-art in optical flow estimation given object blur scenes while Brox *et al.*'s contains a similar optimisation framework and numerical scheme to Portz *et al.*'s, and ranks in the midfield of the Middlebury benchmarks based on overall average. Note that all three baseline methods are evaluated using their default parameters setting; all experiments are performed using a 2.9Ghz Xeon 8-cores, NVIDIA Quadro FX 580, 16Gb memory computer.

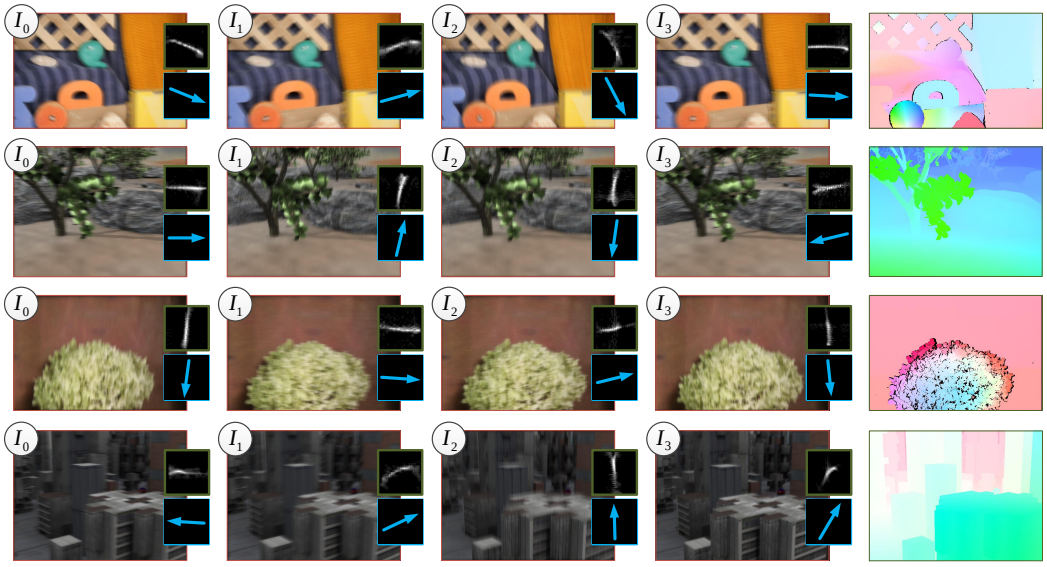


Figure 4-4: The synthetic blur sequences with the blur kernel, tracked camera motion direction and ground truth flow fields. **From Top To Bottom**: sequences of *RubberWhale*, *Urban2*, *Hydrangea* and *Urban2*.

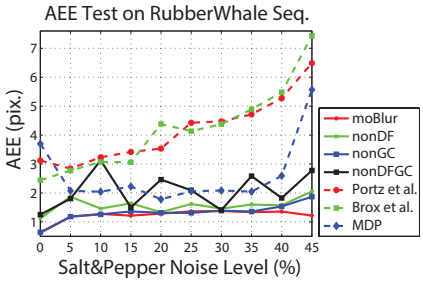
In the following subsections, we compare our algorithm (*moBlur*) and three different implementations (*nonGC*, *nonDF* and *nonGCDF*) against the baseline methods. *nonGC* represents the implementation **without** the *Gradient Constancy* term while *nonDF* denotes an implementation **without** the directional filtering process. *nonGCDF* is the implementation with neither of these features. The results show that our *Blur-Robust Optical Flow Energy* and *Directional High-pass Filter* significantly improve algorithm performance for blur scenes in both synthetic and real-world cases.

4.7.1 Middlebury Dataset with camera shake blur

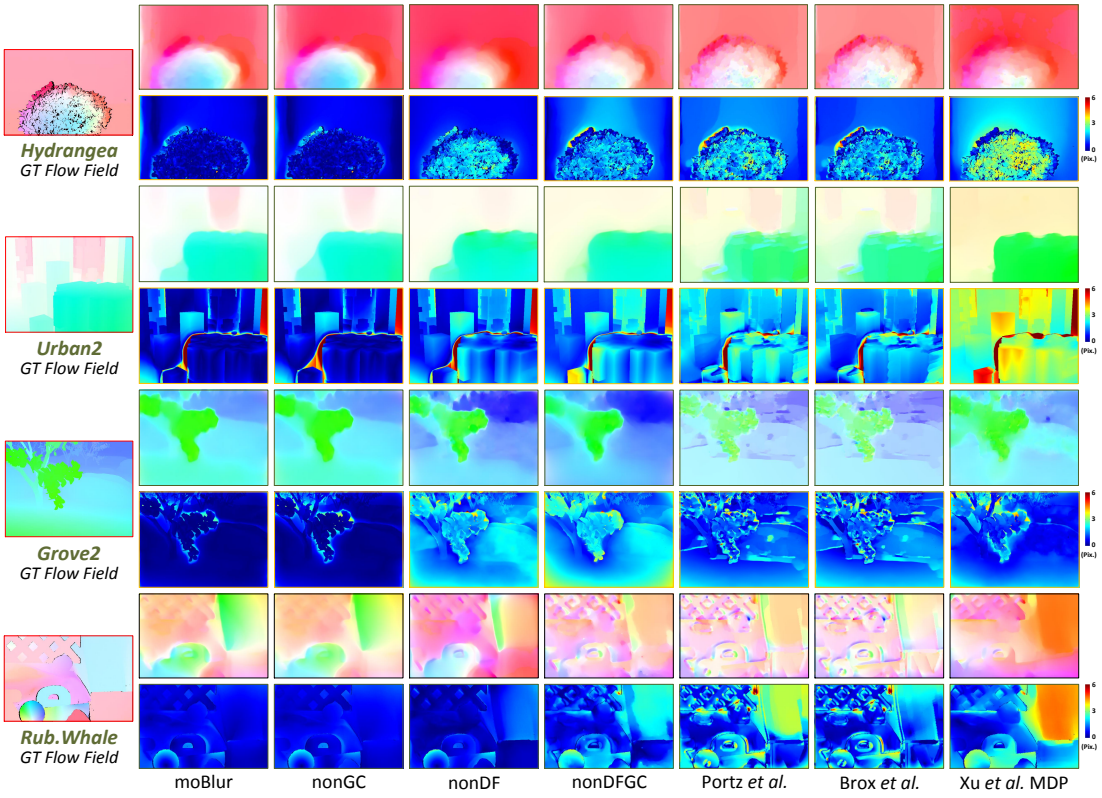
One advance for evaluating optical flow given scenes with object blur is proposed by Portz *et al.* [101] where synthetic *Ground Truth* (GT) scenes are rendered with blurry moving objects against a blur-free static/fixed background. However, their use of synthetic images and controlled object trajectories lead to a lack of global camera shake blur, natural photographic properties and real camera motion behaviour. To overcome these limitations, we render four sequences with camera shake blur and corresponding GT flow-fields by combining sequences from the Middlebury dataset [5] with blur kernels estimated using our system.

In our experiments we select the sequences *Grove2*, *Hydrangea*, *RubberWhale* and *Urban2* from the Middlebury dataset. For each of them, four adjacent frames are selected as latent images along with the GT flow field $\mathbf{w}_{gt}$ (supplied by Middlebury) for the middle pair. 40×40 blur kernels are then estimated [28] from real-world video streams captured using our *RGB-Motion Imaging System*. As shown in Fig. 4-4, those kernels are applied to generate blurry images denoted by I_0, I_1, I_2 and I_3 while the

AEE/AAE	Time Cost	Grove2		Hydrangea		Rub.Whale		Urban2	
		AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE
Ours, moBlur	45	0.47 1	2.34 1	0.67 1	2.19 1	0.62 1	3.67 1	1.36 1	2.87 1
moBlur-nonGC	33	0.49 2	2.38 2	0.95 2	2.23 2	0.64 2	3.71 2	1.54 2	3.03 2
moBlur-nonDF	39	1.52 6	4.96 6	1.83 3	3.00 3	1.12 3	6.45 3	2.50 3	5.19 5
nonDFGC	28	1.62 7	5.14 7	2.28 5	3.21 4	1.25 4	7.71 4	2.98 5	5.44 6
Portz *et al.*	85	1.14 4	4.11 4	2.62 6	3.55 6	3.12 6	8.18 6	3.44 6	5.10 4
Brox *et al.*	27	1.24 5	4.53 5	2.26 4	3.47 5	2.44 5	7.98 5	2.92 4	4.60 3
Xu *et al.*, MDP	466	1.06 3	3.46 3	3.40 7	3.55 6	3.70 6	8.21 7	5.62 7	6.82 7



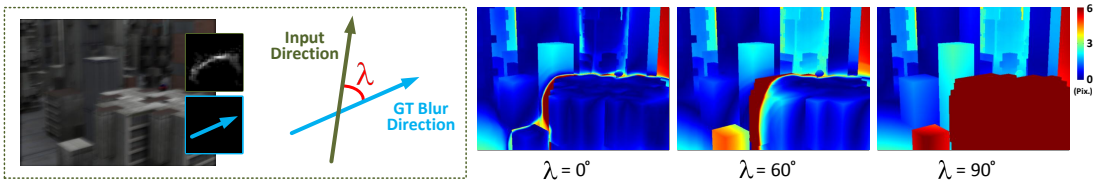
(a) **Left**: Quantitative *Average Endpoint Error* (AEE), *Average Angle Error* (AAE) and *Time Cost* (in second) comparisons on our synthetic sequences where the subscripts show the rank in relative terms. **Right**: AEE measure on *RubberWhale* by ramping up the noise distribution.



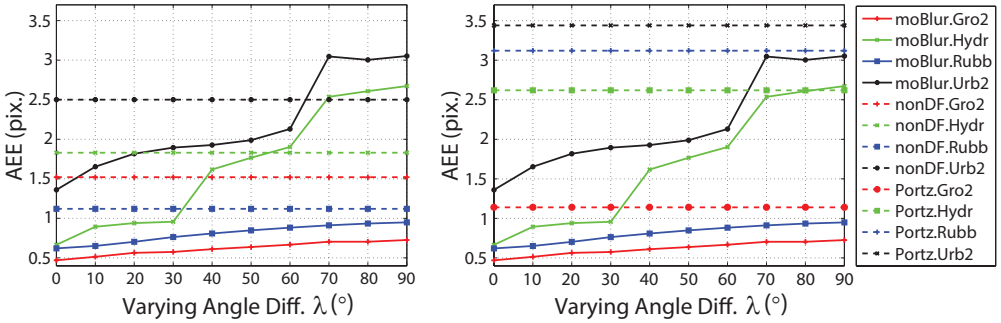
(b) Visual comparison on sequences *RubberWhale*, *Urban2*, *Hydrangea* and *Urban2* by varying baseline methods. For each sequence, **First Row**: optical flow fields from different methods. **Second Row**: the error maps against the ground truth.

Figure 4-5: Quantitative evaluation on four synthetic blur sequences with both camera motion and ground truth.

camera motion direction is set for each frame based on the 3D motion data. Although the $\mathbf{w}_{gt}$ between latent images can be utilised for the evaluation on relative blur images I_* [23, 150], strong blur can significantly violate the original image intensity, which leads to a multiple correspondences problem: a point in the current image corresponds to multiple points in the consecutive image. To remove such multiple correspondences, we sample reasonable correspondence set $\{\hat{\mathbf{w}} \mid \hat{\mathbf{w}} \subset \mathbf{w}_{gt}, |I_2(\mathbf{x}+\hat{\mathbf{w}}) - I_1(\mathbf{x})| < \epsilon\}$ to



(a) sample sequence and error maps



(b) Comparion of on/off filter (c) Comparion to Portz *et al.*

Figure 4-6: AEE measure of our method (*moBlur*) by varying the input motion directions. **(a)**: the overall measure strategy and error maps of *moBlur* on sequence *Urban2*. **(b)**: the quantitative comparison of *moBlur* against *nonDF* by ramping up the angle difference λ. **(c)**: the measure of *moBlur* against Portz *et al.* [101].

use as the GT for the blur images I_* where ϵ denotes a predefined threshold. Once we obtain $\hat{\mathbf{w}}$, both *Average Endpoint Error* (AEE) and *Average Angle Error* (AAE) tests [5] are considered in our evaluation. The computation is formulated as follows:

$$AEE = \frac{1}{n}\sum_{\mathbf{x}}\sqrt{(u-\hat{u})^2+(v-\hat{v})^2} \tag{4.18}$$

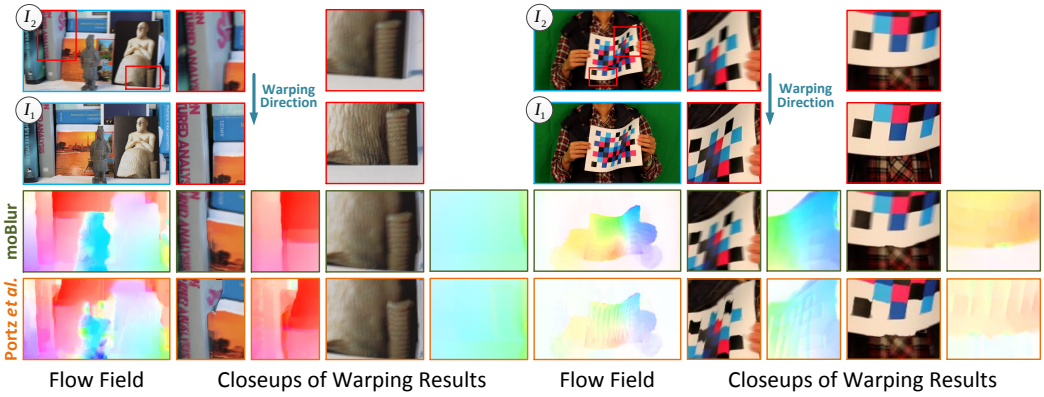
$$AAE = \frac{1}{n}\sum_{\mathbf{x}}\cos^{-1}\left(\frac{1.0+u\times\hat{u}+v\times\hat{v}}{\sqrt{1.0+u^2+v^2}\sqrt{1.0+\hat{u}^2+\hat{v}^2}}\right) \tag{4.19}$$

where $\mathbf{w}=(u,v)^T$ and $\hat{\mathbf{w}}=(\hat{u},\hat{v})^T$ denotes the baseline flow field and the ground truth flow field (by removing multiple correspondences) respectively while n presents the number of ground truth vectors in $\hat{\mathbf{w}}$. The factor 1.0 in AAE is an arbitrary scaling constant to convert the units from pixels to degrees [5]. Fig. 4-5(a) *Left* shows AEE (in pixel) and AAE (in degree) tests on our four synthetic sequences. *moBlur* and *nonGC* lead both AEE and AAE tests in all the trials. Both Brox *et al.* and MDP yield significant error in *Hydrangea*, *RubberWhale* and *Urban2* because those sequences contain large textureless regions with blur, which in turn weakens the inner motion estimation process as shown in Fig. 4-5(b). Fig. 4-5(a) also illustrates the average time cost (second per frame) of the baseline methods. Our method gives reasonable performance (45 sec. per frame) comparing to the state-of-the-art Portz

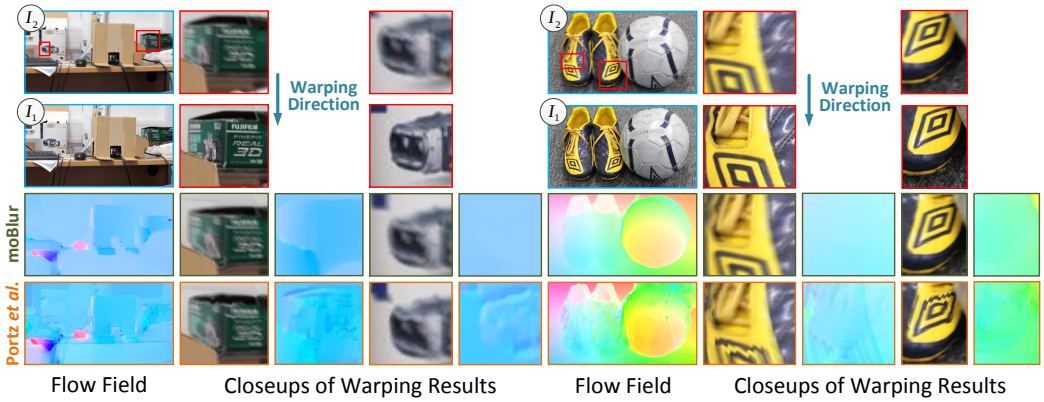
Figure 4-7: The real-world sequences captured along the tracked camera motion. **From Top To Bottom**: sequences of *warrior*, *chessboard*, *LabDesk* and *shoes*.

et al. and MDP even an inner image deblurring process is involved. Furthermore, Fig 4-5(a) *Right* shows the AEE metric for *RubberWhale* by varying the distribution of *Salt&Pepper* noise. It is observed that a higher noise level leads to additional errors for all the baseline methods. Both *moBlur* and *nonGC* yield the best performance while Portz *et al.* and Brox *et al.* show a similar rising AEE trend when the noise increases.

In practice, the system may be used in some challenge scenes, e.g. fast camera shaking, super high frame rate capture, or even infrared interference, etc. In those cases, the wrong tracked camera motion may be given to some specific frames. To investigate how the tracked camera motion affects the accuracy of our algorithm, we compare *moBlur* to *nonDF* (our method without directional filtering) and Portz *et al.* by varying the direction of input camera motion. As shown in Fig. 4-6(a), we rotate the input camera motion vector with respect to the GT blur direction by an angle of λ degrees. Here $\lambda = 0$ represents the ideal situation where the input camera motion has the same direction as the blur direction. The increasing λ simulates more errors in the camera motion estimation. Fig. 4-6(b,c) shows the AEE metric by increasing the λ. We observe that the AEE increases during this test. *moBlur* outperforms the *nonDF* (*moBlur* without the directional filter) in both *Grove2* and *RubberWhale* while *nonDF* provides higher performance in *Hydrangea* when λ is larger than 50°. In addition, *moBlur* outperforms Portz *et al.* in all trials except *Hydrangea* where Portz *et al.* shows a minor advantage (AEE 0.05) when $\lambda = 90°$. The rationale behind this experiment is that the wrong camera motion may yield significant information loss in the directional high-pass filtering. Such information loss harms the deblurring process and consequently leads to errors in the optical flow estimation. Thus, obtaining precise camera motion is the essential part of this system, as well as a potential future research.



(a) Visual comparison on real-world sequences of *warrior* and *chessboard*.



(b) Visual comparison on real-world sequences of *LabDesk* and *shoes*.

Figure 4-8: Visual comparison of image warping on real-world sequences of *warrior*, *chessboard*, *LabDesk* and *shoes*, captured by our *RGB-Motion Imaging System*.

4.7.2 Real-world Dataset

To evaluate our method in the real-world scenes, we capture four sequences *warrior*, *chessboard*, *LabDesk* and *shoes* with tracked camera motion using our *RGB-Motion Imaging System*. As shown in Fig. 4-7, both *warrior* and *chessboard* contain occlusions, large displacements and depth change while the sequences of *LabDesk* and *shoes* embodies the object motion blur and large textureless regions within the same scene. Fig. 4-8 shows visual comparison of our method *moBlur* against Portz *et al.* on these real-world sequences. It is observed that our method preserves appearance details on the object surface and reduce boundary distortion after warping using the flow field. In addition, our method shows robustness given cases where multiple types of blur exist in the same scene (Fig.4-8(b), sequence *shoes*).

4.8 Conclusion

In this chapter, we proposed a hybrid optical flow model by interleaving iterative blind deconvolution and a warping based minimisation scheme. We also highlighted the benefits of both the RGB-Motion data and directional filters in the image deblurring task. Our evaluation demonstrated the high performance of our method against large camera shake blur in both noisy and real-world cases. One limitation in our method is that the spatial invariance assumption for the blur is not valid in some real-world scenes, which may reduce accuracy in the case where the object depth significantly changes. Finding a depth-dependent deconvolution is a challenge for future work.

The related publication is shown as follows:

[75] **W. Li**, Y. Chen, J. Lee, G. Ren, and D. Cosker, *Robust Optical Flow Estimation for Continuous Blurred Scenes using RGB-Motion Imaging and Directional Filtering*, in Proceeding of IEEE Winter Conference on Application of Computer Vision (WACV'14), 2014. (awarded as **Best Student Paper**)

Chapter 5

Dense Nonrigid Tracking in Long Sequences

Tracking nonrigid surface through long image sequences is a fundamental research issue in computer vision. This task relies on estimating correspondences between image pairs over time where error accumulation in tracking can result in *drift*. In this chapter, we propose an optimisation framework with a novel *Anchor Patch* based algorithm which significantly reduces overall tracking errors given long sequences containing nonrigidly deformable objects. The framework may be applied to any tracking algorithm that calculates dense correspondences between images, e.g. optical flow. We demonstrate the success of our approach by showing significant tracking error reduction using 6 existing optical flow algorithms applied to a range of nonrigid benchmarks. We also provide quantitative analysis of our approach given synthetic occlusions and image noise.

5.1 Introduction

Tracking a set of landmark points through multiple images is a fundamental research issue in computer vision. We define tracking in this chapter as the estimation of corresponding sets of vertices, pixels or landmark points between a reference frame and any other frame in the same image sequence. In the last two decades, optical flow has become a popular approach for tracking through image sequences [34, 14]. Compared with feature matching methods e.g. [84], optical flow provides subpixel accuracy and dense correspondence between a pair of images. In this chapter, we focus in particular on improving tracking in image sequences using optical flow, and our contribution applies to this class of algorithm.

One of the main drawbacks of optical flow is *drift* [21]. Errors accumulated between frames over time results in movement away from the correct tracking trajectory.

Between single image pairs, this problem may not be noticeable. However, accumulation when tracking across long sequences can be particularly problematic. Several authors have previously attempted to reduce optical flow *drift* in tracking. DeCarlo *et al.* [34] introduce contour information on a human face to improve tracking stability, while Borshukov *et al.* [14] employ manual correction. More recently, Bradley *et al.* [16] proposed an optimisation method constrained by additional tracking information from multiview video sequences. Beeler *et al.* [8] then introduced the concept of anchor frames for human face tracking. In this approach, the sequence is decomposed into several clips based on anchor images which are visually similar to a reference frame. Their optimisation method shortens the tracking distance from reference frames to the target frame to help alleviate errors. However, their approach is domain specific (faces), and assumes that the entire face will return to a neutral expression (the anchor) several times throughout the sequence. In general, it is difficult to label anchor frames on general object sequences with large displacement motion e.g. waving cloth, as there is usually significant deformation between the reference frame and the other frames. In addition, repeated patterns are typically not global as observed in a face (return to a neutral expression). Rather, they occur in smaller local regions at intermittent intervals.

In this chapter, we focus on tracking long video sequences using optical flow algorithms, and specifically concentrate on reducing *drift*. The general strategy of our approach is to shorten tracking distances for local regions throughout a long sequence. Our proposed framework combines long term feature matching with optical flow estimation. It may be applied to the tracking of general objects with large displacement motion, and results in a significant reduction in *drift*. We first detect *Anchor Frames* for a sequence (Sec. 5.4). This provides an initial set of start points for tracking the sequence. Our main contribution is extending this approach by proposing the concept of *Anchor Patches* (Sec. 5.5). These are corresponding points and patches throughout the sequence which are propagated directly from the reference frame. Our framework substantially reduces overall drift on a tracked image sequence, and may be applied to any optical flow algorithm in a straightforward manner. In our evaluation, we apply the proposed optimisation framework on 6 popular optical flow estimation algorithms to illustrate it's applicability. We provide analysis of our method using 6 synthetic benchmark sequences (Sec. 5.7) generated using a method similar to [49], three of which are degraded by adding occlusion, gaussian noise and salt&pepper noise. In addition, we show its applicability on a popular publicly available real world facial sequence with manually annotated ground truth. We show that our proposed optimisation framework significantly improves tracking accuracy and reduces overall drift when compared against the baseline optical flow approaches alone.

This chapter is organized as follows: In Sec. 5.2, an overview of our proposed

Input: A reference frame, a triangle mesh and an image sequence
1. Computing Optical flow fields (Sec. 5.3)
  1.1 Compute optical flow fields in both forward ($\mathbf{w}_{i\to i+1}$) and backward ($\mathbf{w}'_{i+1\to i}$)
  1.2 Define the *Error Score* function
2. Detect anchor frames and propagate the entire mesh to these frames (Sec. 5.4)
  2.1 Match SIFT features from the reference to every other frame
  2.2 Compute the general *Error Score* on matchings
  2.3 Label the anchor frames from any frame with the low general *Error Score*
3. Label anchor patches on non-anchor frames (Sec. 5.5)
  3.1 Reuse the SIFT feature matching from **2.1**
  3.2 Propagate patches from the reference using *Barycentric Coordinate Mapping*
4. Track remaining patches from anchor frames to non-anchor frames (Sec. 5.6)
  4.1 Propagate patches from the reference to anchor frames (Sec. 5.6.1)
    4.1.1 Compute concatenating optical flow field $\mathbf{w}_{R\to A}$
    4.1.2 Propagate patches from anchor frames to non-anchor frames using $\mathbf{w}_{R\to A}$
    4.1.3 Refine the patches using *Error Score*
  4.2 Propagate patches from anchor frames to non-anchor frames (Sec. 5.6.2)
    4.2.1 Track the patches from the reference to frame i using $\mathbf{w}_{A\to i}$
    4.2.2 Track the patches from the *Nearest Anchor Patches* to frame i
    4.2.3 Eliminate the vertex position conflicts between **4.2.1** and **4.2.2**
Output: A mesh tracked throughout the entire image sequence

Table 5.1: The major steps of the *Anchor Patch* optimisation framework.

optimisation framework is outlined. Sec. 5.3, 5.4, 5.5 and 5.6 give details of the four major steps in our framework. In Sec. 5.7, we evaluate our approach using 6 optical flow algorithms tested on 6 synthetic benchmark sequences and a real world facial sequence.

5.2 System Overview

Our proposed optimisation framework reduces overall optical flow drift given long image sequences, and provides additional robustness against other issues such as large displacements and occlusions. The major procedure is shown in Table 5.1.The aim of our *Anchor Patch optimisation Framework* (APO) is accurately tracking a mesh denoted by $M_R = (V_R, E_R, F_R)$ from a reference frame I_R to every other frame I_i in the sequence. $M_i = (V_i, E_i, F_i)$ denotes the corresponding mesh on frame I_i. In the following sections, the four major steps are discussed in detail.

5.3 Step One: Computing Optical Flow Fields

The first step is to compute an optical flow field between every frame and its successor over a long video sequence in both forward and backward directions (Fig. 5-1). In our evaluation, we consider application of our APO framework on a number of dense correspondence optical flow or tracking approaches, e.g. Brox *et al.* [21], Classic+NL [126]

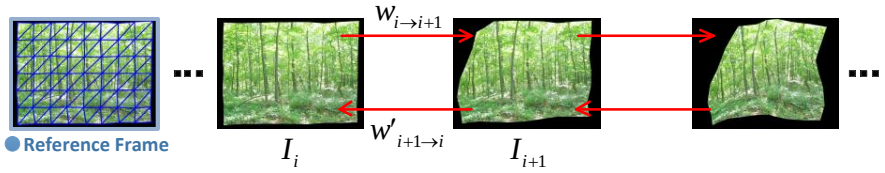


Figure 5-1: Step One. The optical flow fields are computed in both forward ($\mathbf{w}_{i\to i+1}$) and backward ($\mathbf{w}'_{i+1\to i}$) directions between every adjacent images pair in the sequence where the first frame is labelled as a reference frame.

and ITV-L1 [143]. Let $\mathbf{w}_{i\to i+1}$ denote the optical flow field from frame I_i to frame I_{i+1}. Similarly we have $\mathbf{w}'_{i+1\to i}$ denoting the optical flow field from frame I_{i+1} to frame I_i in the backward direction. The optical flow field between frame I_i and I_j where $i < j$ (Forward direction), is denoted by $\mathbf{w}_{i\to j}$ as $\mathbf{w}_{i\to j} = \sum_{i<j} \mathbf{w}_{i\to i+1}$. Similarly, The optical flow field between frame I_j and I_i where $i < j$ (Backward direction), is denoted by $\mathbf{w}'_{j\to i}$ as $\mathbf{w}'_{j\to i} = \sum_{j>i} \mathbf{w}'_{j\to j-1}$.

In order to evaluate the optical flow at a specific pixel $\mathbf{x} = (x, y)^T$, an *Error Score* $E(w)$ from Eq. (3.6) (Sec. 3.3.2) is extended here, where $w = (u, v)^T$ is the optical flow vector at pixel $\mathbf{x}$. The pixel $\mathbf{x}$ in frame I_i is matched to pixel $\mathbf{x}' = (x', y')^T$ in frame I_{i+1} where $\mathbf{x}' = \mathbf{x} + w$. The *Error Score* $E(w)$ is calculated as the weighted *Root Mean Square* (RMS) error at a 3×3 pixel area centred on pixel $\mathbf{x}$ and $\mathbf{x}'$.

$$
\begin{aligned}
E(w) &= \sqrt{\frac{\alpha_1 d(x,y) + \alpha_2 d_{cross}(x,y) + \alpha_3 d_{diag}(x,y)}{\alpha_1 + \alpha_2 + \alpha_3}} \\
d_{diag}(x,y) &= d(x-1,y-1) + d(x+1,y+1) \\
&+ d(x-1,y+1) + d(x+1,y-1) \\
d_{cross}(x,y) &= d(x-1,y) + d(x+1,y) + d(x,y-1) + d(x,y+1) \\
d(x,y) &= |I_i(x,y) - I_{i+1}(x+u, y+v)|^2
\end{aligned} \tag{5.1}
$$

Where α_1, α_2 and α_3 are weights for controlling the contribution of each pixel in the 3×3 area. In our experiments, all these weights are set as $\alpha_1 = 1$, $\alpha_2 = 0.25$ and $\alpha_3 = 0.125$ which refer to the distance from the centre pixel $\mathbf{x}$ of the area. This *Error Score* is intended to evaluate the optical flow at a specific pixel. We also use it to evaluate feature matching scores later in our framework.

5.4 Step Two: Labeling Anchor Frames

After obtaining our optical flow fields, anchor frames are then detected in a similar manner to Beeler *et al.* [8], with the difference that we employ *SIFT* for feature matching as opposed to *Normalised Cross Correlation* (NCC), and additionally use our *Error Score*

Figure 5-2: Step Two. The frames are detected as anchor frames (Red) because of the similar appearance to the reference (Blue). These anchor frames partition the entire sequence into several independent clips which allows tracking performing in parallel.

function (Sec. 5.3) to evaluate matches. The main procedure is as follows (Fig. 5-2):

- **Feature Capture.** A set of *SIFT* features S_R is detected in the reference frame I_R. Note that other features could be employed, but we select *SIFT* due to the general high accuracy and robustness. Here we apply the GPU version matching approach [46] to perform correspondence matching of *SIFT* feature sets S_R to feature set S_i of any other frames I_i.

- **Outlier Rejection.** The aim of this selection process is removing outliers from our feature matching on all the frames. Correspondence matches of the *SIFT* feature set S_R between the reference frame I_R and the target frame I_i are performed. We select the matches which meet $|\mathbf{x}-\mathbf{x}'| < \tau$ where $\mathbf{x}$ is feature position in I_R, $\{\mathbf{x} \in S_R, \mathbf{x} = (x, y)^T\}$; $\mathbf{x}'$ is the corresponding feature position in I_i; τ is a threshold which is set as 30 pixels in our experiments. We find this simple outlier rejection strategy sufficient for most of cases in our experiments (Sec. 5.7). More sophisticated outlier rejection method such as [100] could also be employed.

- **General Error Score.** The general error score is computed for every image as the average of the overall *Error Score* $E(w)$ (Eq. (5.1)). Frames that contain the lowest general error score (below a specific threshold) are selected as anchor frames denoted I_A and the other frames are non-anchor frames. It is because that the general error score is supposed to quantise the general appearance deformation where low score presents the small appearance change. Fig. 5-3 shows this process on our *Carton* benchmark sequence.

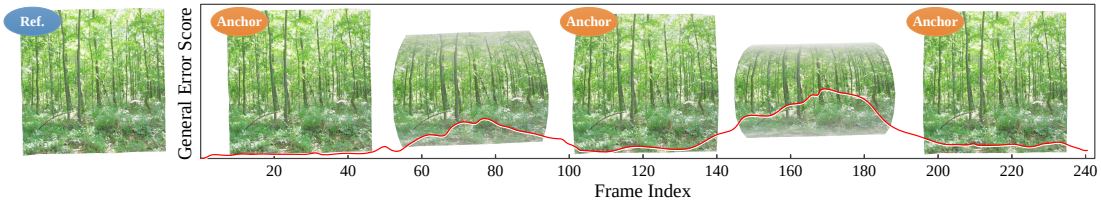


Figure 5-3: The anchor frames are selected based on our general error score which is computed by comparing the reference frame to every other frame in our *Carton* benchmark sequence.

After labeling anchor frames that are visually similar to reference frame, these are used as a basis to partition the entire image sequence into several independent *clips*. This also allows computation in the next steps to be performed in parallel. In addition, the mesh M_R is propagated from the reference frame I_R to each anchor frame I_A using *SIFT* matches and a direct optical flow field between them. More detail can be found in Sec. 5.6.1. The propagated mesh in an anchor frame is denoted $M_A = (V_A, E_A, F_A)$. Because of large displacement motion between anchor frames, and the fact that many images in a deformable sequence may not return to a reference point, these alone are typically insufficient to provide reliable tracking. In the next section, the *Anchor Patch* concept will be introduced to overcome this issue.

5.5 Step Three: Labeling Anchor Patches

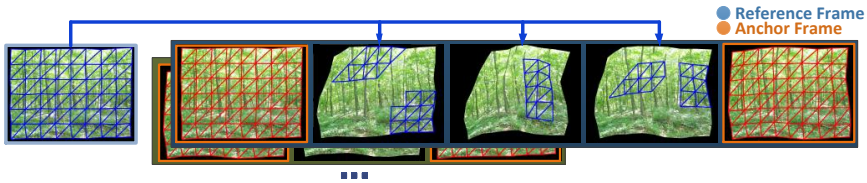


Figure 5-4: Step Three. Anchor patches (blue patches) are label on non-anchor frames within every clip using *SIFT* feature matching and *Barycentric Coordinate Mapping* between reference frame and non-anchor frame.

The motivation of the original *Anchor Frame* method [8] is to provide multiple *Starting Points* for tracking. Since error accumulates, the technique is intended to reduce overall error accumulation across long image sequences. However, as mentioned in the previous section, large displacement motion and complex motion may yield a fact that most images in a video sequence have significant visual differences from the reference frame.

The main observation in long image tracking is that local spatial patterns throughout a sequence may be repeated - i.e. part of a cloth might return to the same position several times throughout a video. We take advantage of these repeating regions in order to track between shorter segments, and thus alleviate error accumulation. Apart from taking an entire image as anchor information, an *Anchor Patch* is defined as a set of individual vertices or a group of pixels in the non-reference frame (any other frame in the sequence), which are highly correspondent to a specific part of the reference. The benefit of using anchor patches is to provide additional information for correcting accumulated errors when tracking using optical flow. This technique can also reduce the impact of a low-quality anchor frame (i.e. the one is too dissimilar from the reference frame). Before anchoring patches on non-anchor frames, we first obtain a set of high-quality *SIFT* feature matches between the reference frame and non-anchor

frames, i.e. those frames are not already labelled as the reference frame, or an existing anchor frame. This process proceeds as follows:

- **Feature Capture.** In order to save the computational time, we reuse the *SIFT* feature sets from *Step Two* (Sec. 5.4). Here the *SIFT* feature set is denoted as S_R in the reference frame I_R; S_i presents a feature set of non-anchor frame I_i.
- **Matching Selection.** We also reuse the refined matchings from *Step Two* (Sec. 5.4). This process generates a matches set $\mathbf{m}_{R\to i}$ from S_R to S_i.

The set of matches $\mathbf{m}_{R\to i}$ is used as our initial basis for anchoring patches on non-anchor frames. In order to obtain final anchor patches, *Barycentric Coordinate Mapping* and *Error Refinement* are applied as follows:

Barycentric Coordinate Mapping

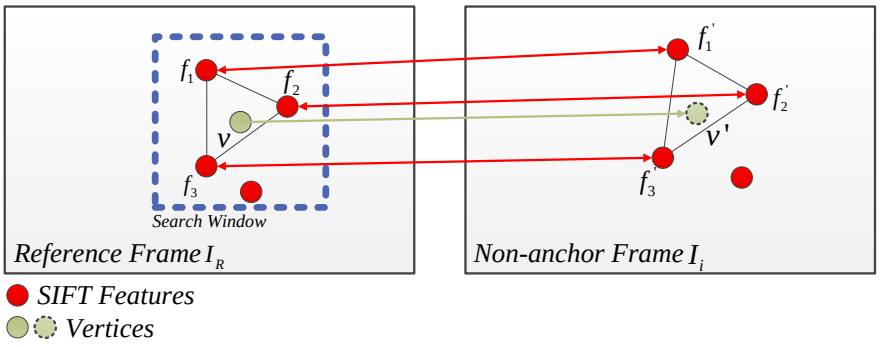


Figure 5-5: Anchoring patches using *Barycentric Coordinate Mapping* and *SIFT* features.

We suppose to determine the pixel position in a non-anchor frame which corresponds to the position of a vertex on the reference mesh M_R in I_R. These correspondences provide our baseline for stable tracking throughout the image sequence. Fig. 5-5 illustrates the process of anchoring patches where $v = (x, y)^T$ denotes a vertex in M_R; $f_* = (x_*, y_*)^T$, and denotes *SIFT* features in the reference frame I_R. Similarly, $f'_* = (x'_*, y'_*)^T$ denotes *SIFT* features in a non-anchor frame I_i. For the non-anchor frame I_i, we have $\{f_k \to f'_k \in \mathbf{m}_{R\to i}, k = 1, 2, 3 \dots\}$ which denotes previously obtained corresponding *SIFT* feature matches. We wish to calculate the new vertex position $v' = (x', y')^T$ in the non-anchor frame I_i. We do this by searching for the three nearest *SIFT* features f_* in a small 5×5 search window centred on the vertex of interest v. Next, v' is calculated by solving the *Barycentric Coordinate Mapping* equations as:

$$\begin{bmatrix} f_1 & f_2 & f_3 \\ f_1' & f_2' & f_3' \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} v \\ v' \end{bmatrix} \tag{5.2}$$

Where β_* are intermediate variables that satisfy $\beta_1 + \beta_2 + \beta_3 = 1$. In practice we found this technique to provide an accurate transformation when applied to small region (5×5 pixel block). However, more sophisticated (although slower) interpolation methods could also be used. The process is performed on every vertex in M_R.

Error Refinement

After *Barycentric Coordinate Mapping*, candidate anchor patches denoted by v_*' are obtained in non-anchor frames I_i. We also have matches $v_* \rightarrow v_*'$, the strength of which can be evaluated using our error equation (5.1). Using this error, we select final anchor patches in a non-anchor frame I_i using $\{P(v_*')|E(v_* \rightarrow v_*') < \eta\}$ where η is a predefined threshold.

5.6 Step Four: Mesh Propagation

The objective of our optimisation framework is to track a mesh M_R from the reference frame to every other frame in an image sequence. Given tracking information from the previous sections, this process is separated into two steps: first, the mesh M_R is propagated from reference frame to anchor frames (Sec. 5.4 and 5.6.1). Second, the propagated mesh M_A is propagated from anchor frames to the non-anchor frames within the clip (Sec. 5.6.2).

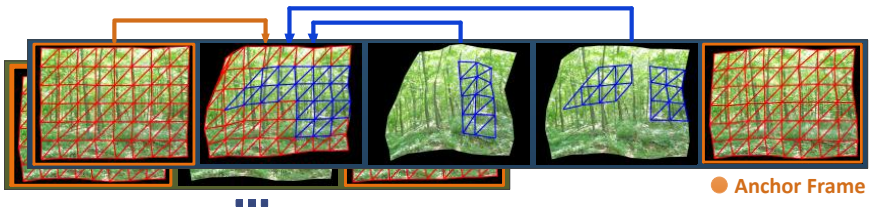


Figure 5-6: Step Four. Tracking other patches from the anchor frame and nearest anchor patches within a clip where the blue patches are anchor patches, selected from *Nearest Anchor Patch*.

5.6.1 Propagating from the reference frame to anchor frames

The mesh propagation process from the reference frame to the anchor frame is as follows:

- **Computing the optical flow field.** The optical flow field $\mathbf{w}_{R\to A}$ directly between the reference frame to the anchor frame is computed by sum up pairwise optical flow fields $\mathbf{w}_{i\to i+1}$ in between.
- **Matching selection.** We propagate the whole mesh M_R from the reference to the anchor frames. For every vertex in M_R, *high error matches* in anchor frames are eliminated (see Error Refinement).
- **Barycentric Coordinate Mapping.** The positions of those eliminated vertices are recomputed by applying *Barycentric Coordinate Mapping* to low error matches. The operation is shown in Fig. 5-5.

After this stage, information for every vertex in M_R is established from the reference frame to the anchor frame.

5.6.2 Propagating from anchor frames to non-anchor frames

The entire image sequence is partitioned into clips which are bound by different anchor frames. The propagation process can be individually performed within these clips in parallel. Within these clips, the anchor patches are supposed to improve overall tracking stability and accuracy. In order to use anchor patches in this process, we define *Nearest Anchor Patch* as follows. For vertex v in M_A, the *Nearest Anchor Patch* of v on frame I_i is the anchor patch $\{v'_{i+k}|v \to v'_{i+k}\}$ on non-anchor frame I_{i+k} which is nearest to I_i in the image sequence. Fig. 5-7 shows an example where frame I_{i+k} is the frame which is nearest to frame I_i in image sequence and contains anchor patch v'_{i+k} matching to v in anchor frame I_A. The main tracking procedure proceeds (Fig. 5-6) as follows:

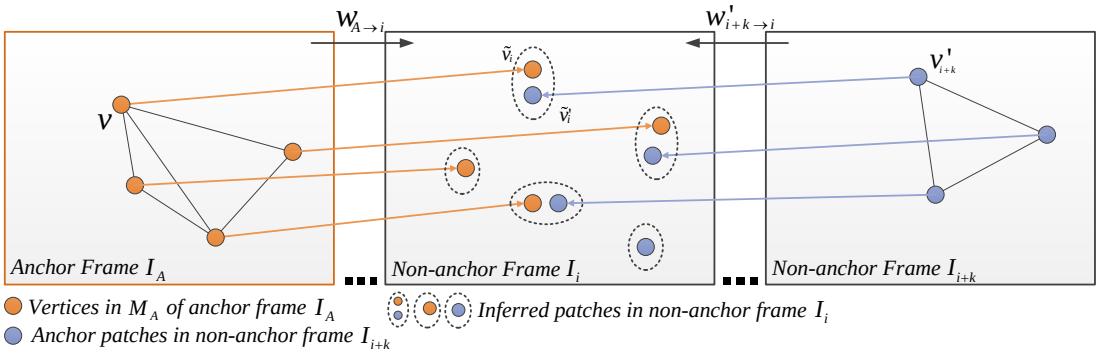


Figure 5-7: Vertex conflict can happen when mesh and anchor patches are propagated to target frame I_i. Here v'_{i+k} is an anchor patch that is strongly matched to v.

- **Mesh propagation.** In order to establish tracking information between anchor frames and non-anchor frame, the mesh M_A is first propagated from anchor frame

I_A to non-anchor frames I_i using the previously calculated optical flow field $\mathbf{w}_{A\to i}$ from *Step One* (Sec. 5.3).

- **Anchor patches propagation.** The *Nearest Anchor Patch* of each vertex v in M_A is searched through the whole clip then propagated to non-anchor frame I_i using the optical flow field in the forward $\mathbf{w}_{*\to i}$ or backward $\mathbf{w}'_{i+k\to i}$ direction.

- **Conflict eliminating.** After propagating the mesh and nearest anchor patches to non-anchor frame I_i, there may be position conflict on some of the propagated vertices. As shown in Fig. 5-7, $\tilde{v}_i$ and $\tilde{v}'_i$ are not in the same desired position. In order to eliminate the conflict, the position of $\{v_i | v \to v_i\}$ matching to v can be calculated using the sum of all weighted candidate positions e.g. $\tilde{v}_i$ and $\tilde{v}'_i$ (Eq.5.3) based on the *Error Score*.

$$v_i = \frac{E(v \to \tilde{v}'_i)\tilde{v}_i + E(v \to \tilde{v}_i)\tilde{v}'_i}{E(v \to \tilde{v}'_i) + E(v \to \tilde{v}_i)} \tag{5.3}$$

Due to the fact that the anchor frames divide the overall sequence into smaller clips, this allows the mesh propagation in between to be calculated in parallel. In the next section we perform an evaluation of our framework.

5.7 Evaluation

We evaluate APO with a range of 6 popular optical flow estimation methods which are publicly available from the *Middlebury Evaluation System* [5]. *Combined local-global Optical Flow* (CLG-TV) [36], *Large Displacement Optical Flow* (LDOF) [21] and Classic+NL [126] are state of the art while the *Horn and Schunck* (HS) [55], *Black and Anandan* (BA) [12, 126], *Improved TV-L1* (ITV-L1) [143] are classic optical flow frameworks and also widely used. CLG-TV is a high speed approach that uses a combination of bilateral filtering and anisotropic regularization and also one of the top three algorithms in the normalized interpolation error test from Middlebury. LDOF is an integration of rich feature descriptors and variational optical flow and one of best current optical flow estimation algorithms for large displacement motion. Classic+NL provides high performance in the Middleburry evaluation by formalizing the median filtering heuristic and Lorentzian penalty as explicit objective functions in an *improved TV-L1* framework. The HS method is a pioneering technique optical flow. BA provides improvements to the HS framework by introducing robust quadratic error formulation. ITV-L1 is a recent and increasingly popular optical flow framework which uses a similar numerical optimisation scheme to Classic+NL. Our choice of a mixture of newer, state of the art methods, with older traditional approaches, is to highlight the fact that irrespective of the approach used, our APO framework provides significantly improved

	Information of the Benchmark Sequences						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
Image Size (pix.)	500 × 500	500 × 500	500 × 500	500 × 500	1024 × 768	1024 × 768	720 × 576
Sequence Length	237	237	237	237	266	307	300
Annotation Points	160	160	160	160	81	63	68
Avg. Feature Amount	364.80	358.32	566.13	1276.50	2498.01	3315.49	2071.11

Table 5.2: An overview of the benchmark sequences in our evaluation. That includes 4 attributes of image size (pixel), sequence length, number of ground truth annotation points per frame and average SIFT feature amount per frame.

tracking in all cases.

For our evaluation, we compare the optical flow estimation methods previously mentioned – with and without our optimisation framework – on 7 long benchmark sequences with ground truth. Table 5.2 gives an overview of the benchmark sequences used in our evaluation. In previous work Garg *et al.* released to the community a set of ground truth data for evaluating optical flow algorithms over long sequences. This is as opposed to the Middlebury dataset, which just considers optical flow between pairs of images, and is therefore not applicable to our framework. The sequences of Garg *et al.* contains 60 frames and are generated using interpolated dense *Motion Capture* (MOCAP) data from real deformations of a waving flag [148]. We use the same MOCAP data to generate a long video sequence and three other degraded sequences, each of which contains 237 frames of size 500 × 500 pixels. The three degraded sequences are generated in order to test the robustness of our APO framework under different image conditions. They are generated by individually adding synthetic occlusions, gaussian noise and salt & pepper noise with the same parameters described in [49]. In order to increase the diversity of the sequences, we include three other sequences. One is a *Talking Face Video* (Frank) sequence which contains 300 frames with 68 ground truth annotation points per frame. The other two are also synthetic benchmark sequences generated using MOCAP data of Salzmann *et al.* [109] from the carton and serviette deformations. One contains 266 frames of size 1024 × 768 while the other contains 307 frames of the same image size. In addition, we also consider the effect of the number of SIFT features detected in the frame, and how this affects overall tracking stability of the APO framework. All optical flow algorithms are applied with default parameter settings from their original papers.

Our baseline optical flow based tracking strategy – for each of the above algorithms – is performed as follows: First, the optical flow field is computed (in forward direction) for every pair of adjacent frames in the sequence. We then mark the initial tracking points in the first frame using the same ground truth points in the same frame of the sequence (Table 5.2). The correspondent points in the next frame are computed based on the optical flow field in between. This process is repeated until correspondent landmark points are obtained in every frame of the sequence. The average *Endpoint Error* (EE) [5] is then calculated against the ground truth annotation points. We then

Methods	Average *Endpoint Error* in pix (AEE)						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [12]	6.14	8.03	11.02	7.79	10.56	5.18	17.57
BA + *APO*	1.72^{2}	1.91^{2}	$\mathbf{7.89}^{1}$	$\mathbf{5.04}^{1}$	2.77	$\mathbf{1.56}^{1}$	6.60
CLG-TV [36]	8.59	10.93	20.28	33.93	28.94	32.17	19.29
CLG-TV + *APO*	2.25	2.97	12.31	18.99	6.95	9.43	7.05
HS [55]	29.16	30.44	29.74	29.43	27.69	37.90	31.27
HS + *APO*	11.68	12.88	17.79	17.21	10.25	10.03	14.19
LDOF [21]	6.21	6.39	16.24	24.14	6.33	5.51	14.73
LDOF + *APO*	1.75^{3}	$\mathbf{1.67}^{1}$	11.65	13.12	$\mathbf{1.18}^{1}$	1.84^{2}	$\mathbf{3.12}^{1}$
Classic+NL [126]	7.07	10.61	12.65	9.50	5.72	6.62	17.32
Classic+NL + *APO*	2.15	3.18	8.31^{2}	6.46^{2}	1.34^{2}	2.03^{3}	3.44^{2}
ITV-L1 [143]	5.73	8.25	17.29	14.49	5.34	7.11	17.91
ITV-L1 + *APO*	$\mathbf{1.50}^{1}$	2.33^{3}	9.53^{3}	7.70^{3}	1.70^{3}	2.36	3.69^{3}

(a) Average *Endpoint Error* (AEE) comparison of different methods with our optimisation framework on the benchmark sequences.

Methods	Average *Endpoint Error* in pix (AEE)						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
Garg *et al.*, PCA [49]	0.61	0.71	1.64	1.21	N/A	N/A	N/A
Garg *et al.*, DCT [49]	0.59	0.74	1.86	1.54	N/A	N/A	N/A
Pizarro *et al.* [100]	0.79	0.81	0.99	0.98	N/A	N/A	N/A

(b) Average *Endpoint Error* (AEE) comparison of Garg *et al.* and Pizarro *et al.* on the benchmark sequences (directly tracking from the reference to any other frames).

Figure 5-8: Average *Endpoint Error* (AEE) comparison on our long benchmark sequences.

apply our APO framework using the same optical flow fields.). Note that the parameter values relevant to the APO framework are initially and experimentally selected, but then remain constant in all our evaluations.

Table 5-8(a) shows the measurement of average *Endpoint Error* (AEE) in pixels over all the frames of the sequences. We highlight the top three best AEE measures for each sequence using superscripts next to different values. Notice that APO significantly reduces the AEE compared to the baseline optical flow methods. Our optimisation framework yields the best AEE measure in all the cases. For instance, *ITV-L1* with APO performs the best in sequence *Original* while *LDOF* with APO yields the best result in sequence *Frank*. We also observe that although in the *Guass.Noise* and *S&P.Noise* sequences the improvement is less than in the unaltered sequences, the overall result is still an improvement with the addition of APO. We also observe that *LDOF* gives good results even without APO. It is because that the *LDOF* framework takes into account both regular optical flow energy and the feature technique. The latter contributes additional accuracy to the final result.

Table 5-8(b) shows another experiment, in which we performs Garg *et al.* [49] and Pizarro *et al.* [100] on our benchmark sequences (results on *Carton*, *Serviette* and *Frank* are not available.) using a direct tracking strategy. Here we compute the optical flow fields directly from the reference to any other frames of the sequence. The annotation points are then directly tracked to the test frames using those flow fields. Note that the numbers in Table 5-8(b) may be slightly different from the similar experiment (Table 3-7(a) in Ch. 3). It is because that, first, the sequences are extended to 237

Methods	Average *Endpoint Error* (AEE) on the First 30 Frames						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [12]	1.57	1.72	3.87	2.71	2.37	1.56^{3}	8.76
BA + *APO*	1.41^{3}	1.65	3.66^{3}	2.13^{2}	2.17	$\mathbf{1.13}^{1}$	5.40
CLG-TV [36]	2.40	2.60	6.71	8.77	8.10	5.54	8.60
CLG-TV + *APO*	2.10	2.24	6.53	8.39	4.79	5.11	7.35
HS [55]	33.67	35.70	35.05	34.50	26.16	22.08	12.76
HS + *APO*	16.11	16.32	13.78	19.37	9.78	6.33	9.19
LDOF [21]	2.38	2.37	3.96	4.03	3.90	2.52	8.51
LDOF + *APO*	1.15^{2}	$\mathbf{0.97}^{1}$	3.75	2.66	$\mathbf{0.89}^{1}$	1.44^{2}	$\mathbf{2.82}^{1}$
Classic+NL [126]	1.63	1.76	3.61^{2}	2.51^{3}	2.18	1.75	8.77
Classic+NL + *APO*	1.51	1.33^{2}	$\mathbf{3.54}^{1}$	$\mathbf{1.99}^{1}$	1.24^{2}	1.68	3.70^{3}
ITV-L1 [143]	1.55	1.76	6.27	5.07	2.37	2.01	9.22
ITV-L1 + *APO*	$\mathbf{0.99}^{1}$	1.31^{2}	5.77	4.65	1.69^{3}	1.71	3.48^{2}

Table 5.3: Average *Endpoint Error* (AEE) comparison of different methods with our optimisation framework on the first 30 frames of the benchmark sequences.

frames which is around 3 times longer than the one in Ch. 3; second, we evaluate the tracking results of only 160 annotation points instead of all the pixel. We observe that both Garg *et al.* and Pizarro *et al.* give higher accuracy than any other baseline method in our experiment. The hidden conditions are (1) the tracking distance is minimum for Garg *et al.* and Pizarro *et al.* which very much reduces the accumulate errors; (2) both Garg *et al.* and Pizarro *et al.* shows high accuracy for nonrigid surface tracking in the record [49, 100]. And all our sequences contain single nonrigid object. However, such direct tracking strategy cannot handle the situation where objects may be temporally out of the scene. In addition, the object appearance in the reference may be significantly different from the one in some other frames of the sequence. That brings extra difficulty to optical flow estimation.

While we concern ourselves primarily with tracking over long sequences, the shorter sequences are consider as well. In Table 5.3, the AEE measures of various methods are compared on the first 30 frames of our benchmark sequences. We observe similar AEE measures as in the long sequence case (Table 5-8). The APO framework significantly increases the tracking accuracy – outperforming the baseline tracking methods in all cases even given degradation (e.g. *Gauss.Noise* and *S&P.Noise*). Moreover, the *BA* with APO is also observed to overfit in the noisy sequences while *Classic+NL* with APO yields the best measures in both sequences of *Gauss.Noise* and *S&P.Noise*.

We also evaluate the effect on tracking accuracy by varying the number of selected features. Different numbers (50% and 0%) of features are randomly selected from the initial full detection feature set before performing *Anchor Patch* detection. Information on our total number of features can be found in Table 5.2, e.g. there are 364.80 features averagely on each frame of the sequence *Original*. Table 5.4 shows an AEE comparison given various numbers of features. We observe that AEE is improved given more features in all cases. Another interesting observation is that our optimisation framework provides lower error against the baseline tracking strategy even given sparse or no features (0% feature). Note that in this case, our APO framework defaults to

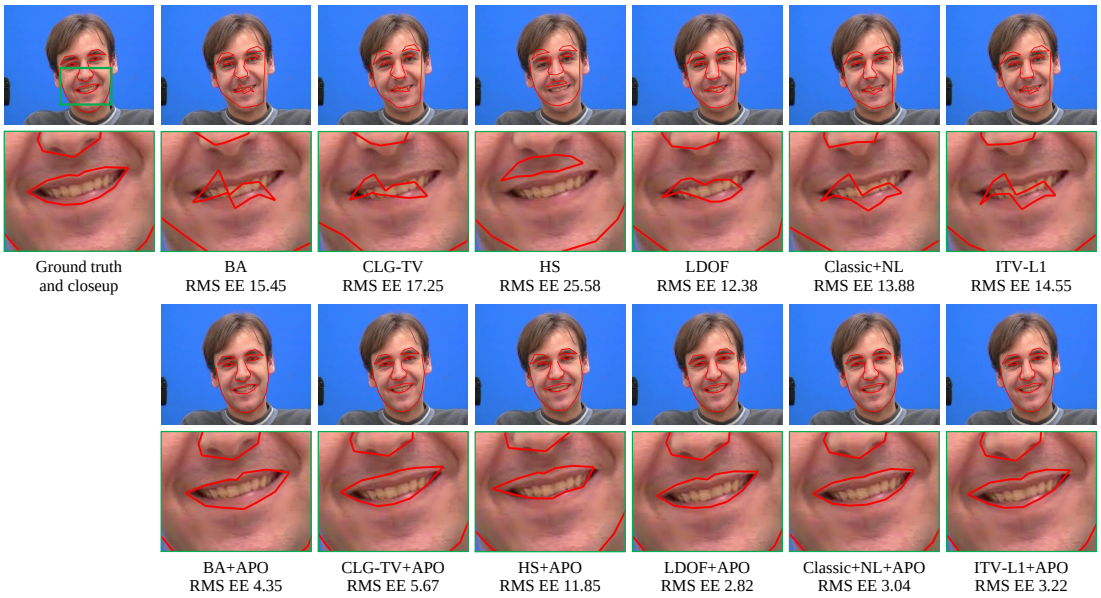
Methods	Average *Endpoint Error* (AEE) on Different Feature Distributions						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [12], **No *APO***	6.14	8.03	11.02	7.79	10.56	5.18	17.57
***APO*, 100% Feature**	1.72[2]	1.91[2]	**7.89**[1]	**5.04**[1]	2.77	**1.56**[1]	6.60
***APO*, 50% Feature**	3.64	4.71	8.06	6.12	5.89	2.98	10.63
***APO*, 0% Feature**	5.12	6.44	9.23	7.21	8.69	4.35	12.69
CLG-TV [36], **No *APO***	8.59	10.93	20.28	33.93	28.94	32.17	19.29
***APO*, 100% Feature**	2.25	2.97	12.31	18.99	6.95	9.43	7.05
***APO*, 50% Feature**	4.86	6.51	14.39	22.72	15.36	19.91	12.00
***APO*, 0% Feature**	6.94	9.11	16.83	26.03	23.57	24.03	15.07
HS [55], **No *APO***	29.16	30.44	29.74	29.43	27.69	37.90	31.27
***APO*, 100% Feature**	11.68	12.88	17.79	17.21	10.25	10.03	14.19
***APO*, 50% Feature**	18.13	20.28	20.66	19.91	17.39	25.99	23.45
***APO*, 0% Feature**	24.73	27.11	23.97	23.40	24.09	33.11	29.17
LDOF [21], **No *APO***	6.21	6.39	16.24	24.14	6.33	5.51	14.73
***APO*, 100% Feature**	1.75[3]	**1.67**[1]	11.65	13.12	**1.18**[1]	1.84[2]	**3.12**[1]
***APO*, 50% Feature**	3.21	3.09	12.18	15.02	2.90	3.74	8.66
***APO*, 0% Feature**	5.08	5.24	14.11	18.46	5.45	4.89	11.76
Classic+NL [126], **No *APO***	7.07	10.61	12.65	9.50	5.72	6.62	17.32
***APO*, 100% Feature**	2.15	3.18	8.31[2]	6.46[2]	1.34[2]	2.03[3]	3.44[2]
***APO*, 50% Feature**	4.00	6.39	9.48	7.33	3.89	4.00	10.14
***APO*, 0% Feature**	5.96	7.78	11.64	8.98	4.78	6.00	13.27
ITV-L1 [143], **No *APO***	5.73	8.25	17.29	14.49	5.34	7.11	17.91
***APO*, 100% Feature**	**1.50**[1]	2.33[3]	9.53[3]	7.70[3]	1.70[3]	2.36	3.69[3]
***APO*, 50% Feature**	3.59	5.17	10.93	8.47	3.41	5.00	10.11
***APO*, 0% Feature**	4.77	6.92	12.50	10.31	4.43	5.95	14.29

Table 5.4: Average *Endpoint Error* (AEE) comparison on the benchmark sequences with varying feature distributions.

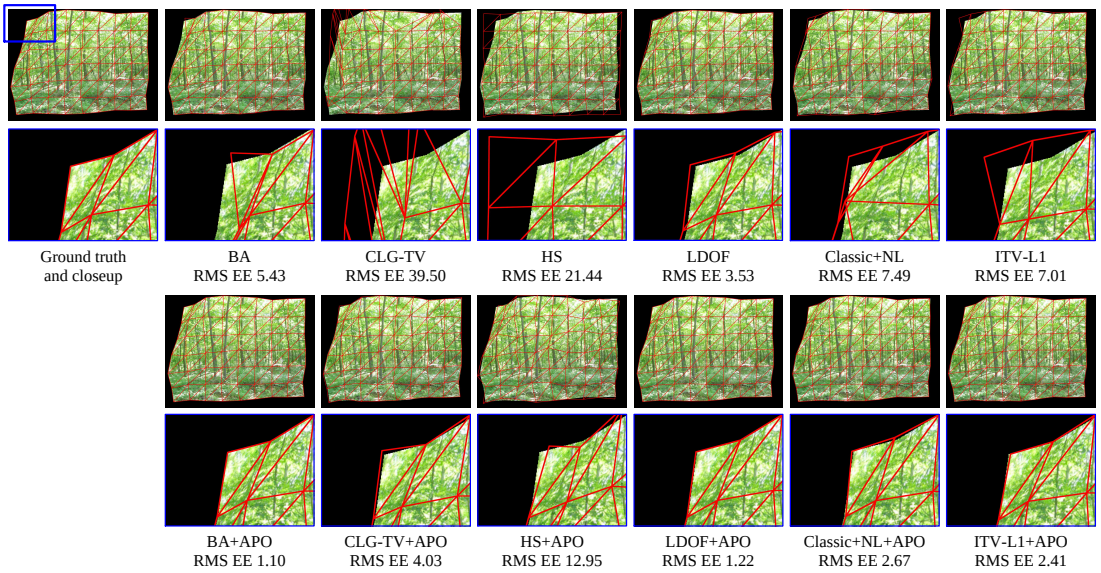
using an optical flow method with just the *Anchor Frame* approach [8]. Also note – for example by comparing to Table 3 – that this indicates that the APO framework also provides significant tracking improvement over using anchor frames alone.

We also make the visual comparisons on two of our sequences, *Frank* and *Serviette*. The former is real world sequence with ground truth annotation points, while the latter is synthetic sequence overlaid with a ground truth mesh. In Fig. 5-9, we observe noticeable *drift* problems given the baseline optical flow tracking strategy. Also note that more details can be found in the corresponding video footage of `http://www.cs.bath.ac.uk/~wl281/apo/APO.mp4` in which we visually show that the APO framework significantly reduces the *drift* problem.

The computational consumption of our framework heavily relies on the supplementary optical flow method, because we need to calculate the optical flow fields twice (forward and backward) for every pair of adjacent images. Apart from this, our framework can be implemented in a parallel computation fashion. Anchor frames divide the sequence into clips which give multiple start points for tracking. In the implementation, a GPU version of SIFT approach [46] is applied for feature detection and matching (around 10 frames per second on our benchmarks). The whole framework is constructed under CUDA platform. Assuming all optical flow fields are obtained, our framework reach real-time efficiency (around 2 frames per second) on our benchmarks using on a 2.9Ghz Xeon 8-cores, NVIDIA Quadro FX 580, 16Gb memory computer.



(a) Visual comparison of different methods on the frame 88 of the sequence *Frank*.



(b) Visual comparison of different methods on the frame 192 of the sequence *Serviette*.

Figure 5-9: Visual comparison and AEE measures on sequences of *Frank* and *Serviette*.

5.8 Conclusion

In this chapter, we have presented an optimisation framework based on *Anchor Patches* for improving mesh or sparse point set tracking during long video image sequences. Our optimisation framework anchors image regions throughout the sequence to mitigate the effect of *Error Accumulation* and *Drift*. In our evaluation, we have compared APO combined with 6 popular optical flow estimation algorithms against baseline tracking on 7 benchmark sequences. This includes 6 synthetic benchmark sequences with real world

deformation and 1 real world sequence. We have demonstrated that APO provides significant tracking improvements for dense tracking on long video sequences than using baseline optical flow tracking alone.

The related publication is shown as follows:

[76] **W. Li**, D. Cosker, and M. Brown, *An Anchor Patch Based Optimisation Framework for Reducing Optical Flow Drift in Long Image Sequences*, in Proceeding of Asian Conference on Computer Vision (ACCV'12), Springer, November 2012, pp. 112–125.

Chapter 6

Dense Ground Truth Capture on Nonrigid Surfaces

In this chapter we present the first ground truth data set of nonrigidly deforming real-world scenes (both long and short video sequences). To construct ground truth for the RGB sequences, we simultaneously capture Near-Infrared (NIR) image sequences where the dense markers – visible only in NIR – represent the ground truth positions, allowing comparison between the RGB tracked positions and the formation of error metrics. Our novel ground truth construction protocol may also be adopted to capture other types of deformable objects, thus opening ground truth opportunities in other difficult-to-track problems. Unlike previous datasets containing nonrigidly deforming sequences using synthetic data, the capture of real-world objects yields realistic photometric effects - such as blur and illumination change - as well as occlusion and complex deformations. A public evaluation website is constructed to allow for ranking of RGB image based optical flow and other dense tracking algorithms, with varying statistical measures. Furthermore, we present the first RGB-NIR multispectral optical flow formulation allowing for overall optimisation of the optical flow energy by maximizing the distinguishing information from both the RGB and the complementary NIR channels. In our experiments we evaluate eight existing optical flow methods on our new dataset, as well as examine our multispectral optical flow algorithm by varying the input channels across RGB, NIR and RGB-NIR.

6.1 Introduction

Multispectral imaging techniques have been widely adopted across computer vision. One particular form of this – *RGB&Near-Infrared* (RGB-NIR) – has recently been shown useful in multispectral SIFT [17], image dehazing [111] and registration [41]. A property of such imaging is the potential to apply markers visible in one spectrum

(e.g. NIR), but invisible in another (e.g. RGB). In this chapter, we employ RGB-NIR imaging combined with *NIR Visible Dyes* and propose a spatio-temporally dense *Ground Truth* (GT) dataset consisting of nonrigid motion from real-world objects and scenes. We also demonstrate the effectiveness of multispectral (RGB-NIR) optical flow by proposing for the first time a method which utilizes information from both channels.

The quantitative evaluation of optical flow algorithms is a difficult challenge – particularly given long nonrigid scenes with natural noise. The *Middlebury* benchmark [5] is currently the most widely used GT in the community, but is limited by its lack of object blur, complex nonrigid motion and long image sequences. Most of these limitations are due to the stop-motion method of capture: a scene is first captured under normal lighting; and then a second image of the same scene is captured using ultraviolet lighting. To address these limitations, Butler *et al.* [23] proposed an optical flow dataset based on a 3D animated film *Sintel*, which contains inter-frame GT through long sequences and geometric blur under different renderings. However their inherent limitation is the use of synthetic sequences, which lacks real-world photometric effects and textural properties. Similar to *Sintel*, Garg *et al.* [49] rendered synthetic video sequences with accompanying GT by projecting the scene motion (*Motion Capture*) of a realistic waving flag onto the image plane.

The variational optical flow model has been extensively studied in the last two decades, beginning with the pioneering work of Horn and Schunck [55] and Lucas and Kanade [87]. Some complementary concepts have since been developed to deal with the shortcomings of their original models such as spatial discontinuities [12], large displacements [21], motion details loss through coarse-to-fine minimisation [152] and local smoothness. Of these methods, Xu *et al.*'s (MDP) [152] approach is currently ranked top (by average) in the *Middlebury* evaluation while our LME approach (Chapter 3) shows the state-of-the-art performance given nonrigid surface motion [49]. However, all of these methods are applied on image pairs within the visible spectrum and are sensitive to motions in large featureless regions in which the basic *Intensity Consistency* assumption is weakened.

6.1.1 Contributions

The major contribution in this chapter is the use of an RGB-NIR imaging system, combined with NIR visible dyes, in order to propose: ***(1)*** a nonrigid optical flow GT dataset and evaluation website containing dense inter-frame correspondences from eight short and five long sequences with varying properties, and ***(2)*** the first multispectral (RGB-NIR) optical flow model (*vnflow*) – which uses the best available image features in either channel to enhance motion analysis.

In our experiments, we evaluate eight existing optical flow methods as well as *vnflow* on our dataset and illustrate the practical benefit of combined RGB-NIR optical flow.

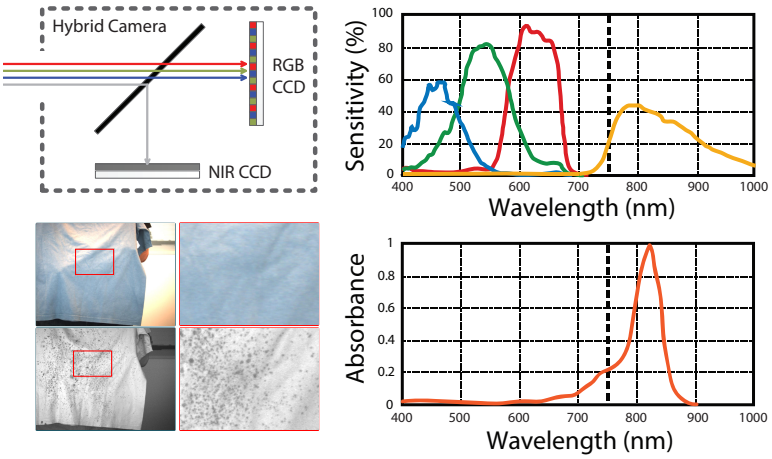


Figure 6-1: RGB-NIR Camera and the NIR visible dyes. **Top Left**: The inside structure of the camera. **Bottom Left**: Sample images captured by the RGB CCD sensor and NIR CCD sensor respectively. **Top Right**: The relative transmittance of RGB CCD sensor and NIR CCD sensor (yellow) respectively. **Bottom Right**: The absorbance of the NIR visible dyes respect to various wavelength.

6.2 RGB-NIR Imaging System

A typical human eye can respond to light with wavelengths in the range of approximately 390 to 700 nm – the visible light spectrum; and usually has sharp sensitivity at around 555 nm. However, current CCD (or CMOS) image sensors utilized in digital cameras are more sensitive and respond to a wider wavelength range between approximately 350 and 1100 nm [17]. This spectrum largely covers the near-infrared (NIR) range, which is approximately 700 to 1400 nm.

RGB-NIR Camera In this work, a hybrid camera (JAI AD-080GE) is used to capture both RGB and NIR images from the same scene simultaneously. Fig. 6-1 shows internal construction of the camera, where natural light is split onto the RGB and NIR CCD sensors respectively. As opposed to experimental bench-based RGB-NIR beam-splitter setups [24], the overall system is both compact and portable (measuring approximately $5 \times 3 \times 3$ inches).

NIR Visible Dyes In order to generate dense features on object surfaces for our GT dataset, we utilize *NIR Visible Dyes* (NIR819D, QCR Solutions Corp) which absorb the spectrum in a range of approximately 700 to 870 nm with a peak at around 819 nm. Fig. 6-1 shows dense patches painted by our dyes is invisible in the NIR channel while remaining invisible in the RGB channel. To illustrate the statistical dependencies of the patches between different bands, $20,000$ RGB-NIR patches (3×3 pix.) with the dyes applied are randomly selected and plotted as pairwise distributions using

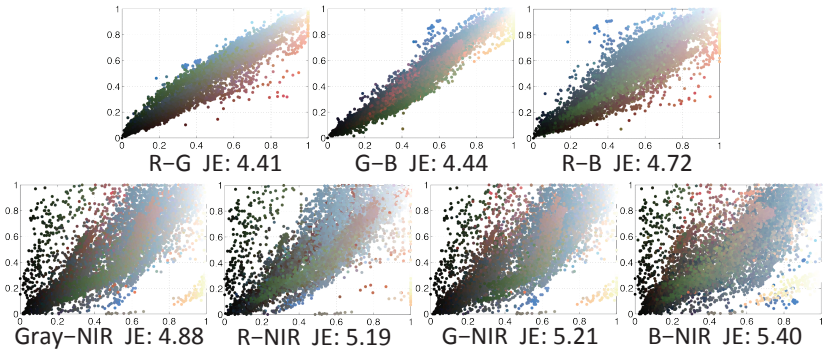


Figure 6-2: Pairwise distributions for the RGB and NIR channels of 20,000 sampled patches from our ground truth dataset.

joint entropy in Fig. 6-2. Note that we compute the joint entropy as $H(X,Y) = -\sum_{X,Y} P(X,Y) log_2[P(X,Y)]$. It is observed that the joint entropy of {R,G,B,Gray}-NIR is larger than between the visible bands (R,G,B). Therefore, the *NIR Visible Dyes* can be used to provide extra visible information in what would usually be a plain textured region in RGB channel.

Motion Control Component To precisely control the displacement of objects in our GT scenes, a motion control mechanism is constructed using LEGO NXT robotics kits which produce controllable and uniform inter-frame movements for our GT surfaces. In the following section, we describe this RGB-NIR dataset, as well as our proposed evaluation methods.

6.3 Dense RGB-NIR Ground Truth Dataset

Ground Truth (GT) for optical flow is difficult to capture. One important advance in this area was proposed by Baker *et al.* with the introduction of the *Middlebury* benchmark [5]. Due to their contribution, the optical flow community has rapidly developed in recent years. However, Baker *et al.* also point out limitations of their work [5], such as a lack of object blur and occluded motion – some of which are discussed in more recent state-of-the-art benchmarks [23]. The main limitations of current benchmarks, which we address in our dataset, are as follows:

Long Image Sequences As discussed in [23], most of the *Middlebury* sequences are short in length. While *Sintel* provides long synthetic sequences (more than 50 frames) and GT for each pair of frames, our dataset provides long sequences from real-world objects – thus exhibiting realistic photometric effects and textural properties.

Realistic Noise The lack of realistic blur is a common issue in both *Middlebury* and *Sintel*. Our dataset includes realistic camera blur and other noise, e.g. strong shadows, reflectance and illumination changes.

Complex Nonrigid Motions Unlike *Middlebury* and *Sintel*, our dataset is specifically focused on nonrigid motion, containing examples of stretching, large bends and creases.

Facilitating the capture of aforementioned properties – with appropriate GT flow fields – is therefore one of the main innovations and contributions of this work. Our dataset contains two types of GT sequences as follows:

Short Sequences Similar to *Middlebury*, we capture eight sequences, each of which contains ten frames with dense GT for the middle image pair. Each sequence is captured so as to include specific common image properties (nonrigid motion, noise, etc).

Long Sequences Five long sequences are captured with dense inter-frame GT for every neighbouring image pair. Each sequence contains 50 frames and is designed to include multiple realistic photometric effects and nonrigid motion.

We next describe the process of GT capture and estimation in detail.

6.3.1 Ground Truth Capture and Estimation

In order to acquire our GT, we construct a controllable scene (i.e. lighting and motion properties) to be captured with the *RGB-NIR Imaging System* mentioned in Sec. 6.2. Our *NIR Visible Dyes* are spread onto object surfaces in order to generate fine patterns of which the diameter is within 1 mm, with a maximum 2 mm distance between any pair of neighbouring patterns. Our *RGB-NIR Camera* simultaneously captures a series of continuous images in both the RGB and NIR channels at 20 FPS.

Image Properties Our *RGB-NIR camera* captures images at 1296×966 pixels. The *Motion Control Component* of our system allows us to precisely range motions from subpixel to 40 pixels. Similar to *Middlebury*, all the captured RGB sequences are downsampled by a factor of 3, resulting in image size of 432×322 after the *Subpixel Motion Estimation* step (will be presented later in this subsection).

Pixel Correspondence The dyes patterns on the object surfaces are small in scale but still highly variable in terms of intensity and shape. Their diameters are generally less than 1 mm, corresponding to approximately 1 pixel of the image (Fig. 6-5(b)). Therefore pixel correspondences are achieved by matching the dyes patterns between

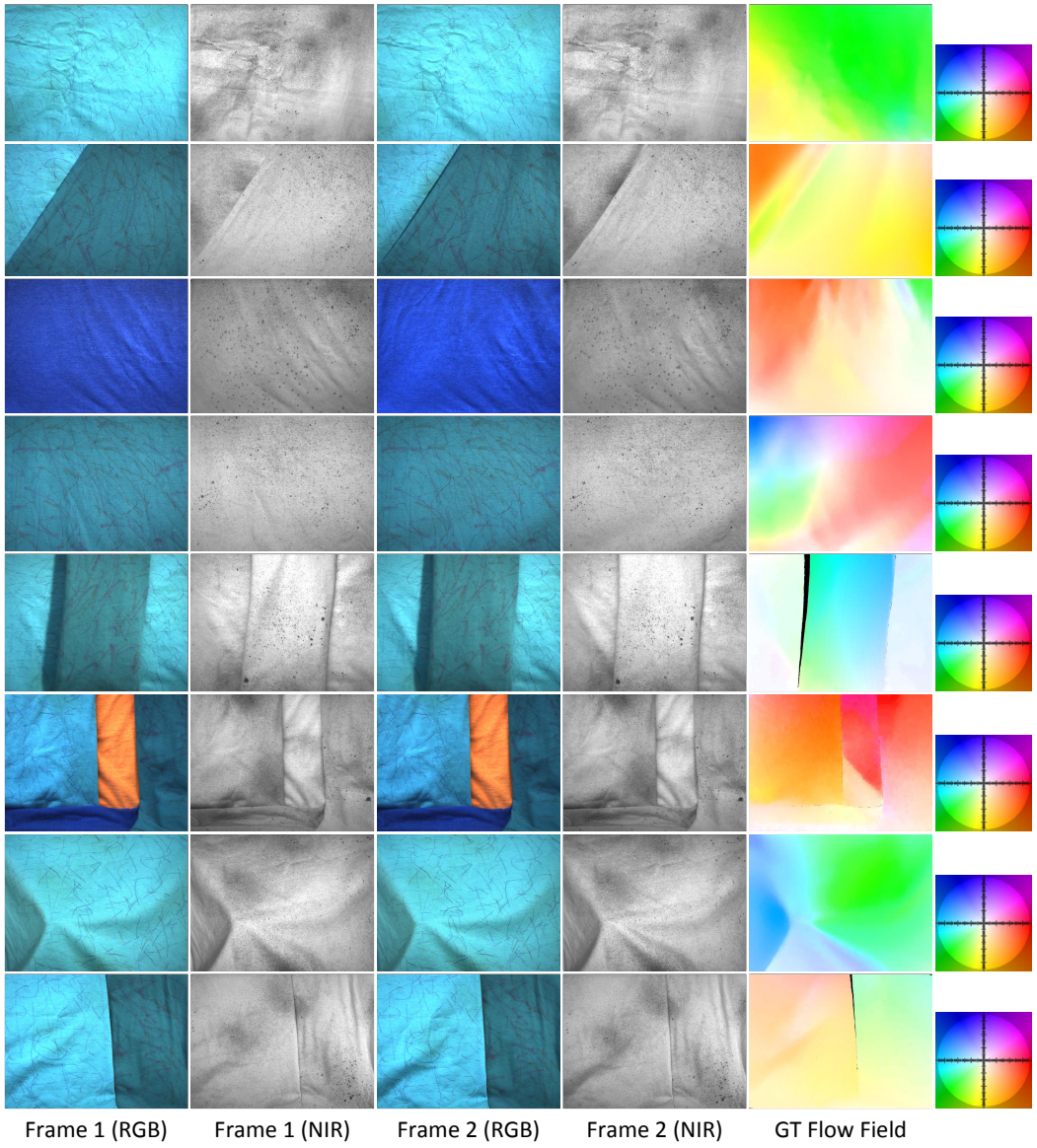


Figure 6-3: The short sequences in our GT dataset. **Top To Bottom**: *illumination*, *mObjs*, *featureless*, *single*, *str.shadow*, *triObjs*, *blur* and *crease*.

neighbouring NIR images. Unlike the Colour-*SSD* tracker used in *Middlebury*, we consider both intensity and shape. A SIFT descriptor with 128 dimensions is computed for each pixel in an image. We nominate a GT match between pixels where the *Euclidean Distance* of their SIFT vectors is smallest within a given search window. This window size is predefined as the maximum motion in the *Motion Control Component*. To improve robustness we examine the matched results across adjacent frames. A correspondence is labelled with a value "*NAN*" (Not-A-Number) if the intensity difference between the forward matched result and the backward matched result is greater than a predefined threshold. The region mask containing "*NAN*" values is recorded as an

occlusion map.

Subpixel Motion Estimation After obtaining GT pixel correspondence, we follow the *Middlebury* subpixel motion estimation process. We apply the *Lucas-Kanade* kernel [87] to each search window for subpixel motion using 1/20 pixel precision. We then calculate the average of up to 9 motion vectors in each 3×3 window in order to downsample the motion field to dimension 432×322.

Realistic Noise The controllable nature of our *RGB-NIR Imaging System* allows us to incorporate varieties of noise and artefacts into our GT dataset. We increase the exposure time of the RGB CCD sensor to bring object blur into the visible channel, while using a suitably fast exposure time on the NIR CCD sensor to capture a corresponding blur-free image. Alternatively, defocus blur could also be obtained by modifying the aperture settings. Shadow and illumination changes are generated using infrared-free light (LED lighting), leading to realistic shadows in the visible channel without affecting illumination in NIR channel (Fig. 6-5(a,b)).

Sequence Descriptions Fig. 6-3 shows all eight short sequences from our dataset. In this short sequence category, *single* refers to nonrigid motion of single object. *illumination* refers to strong reflectance and illumination change while both *mObjs* and *triObjs* contain multiple objects with nonrigid movement. *featureless* contains small motions for a featureless object surface while *crease* presents the large crease on multiple objects. *blur* and *str.shadow* show both camera blur and strong shadow respectively. Furthermore, sample frames of five long sequences is illustrated in Fig. 6-4. In this long sequence category, *mBlur* demonstrates focus blur, motion blur and large displacements, while *circle* contains complex crease motions. *crush* presents a large crush of an object with self occlusions and *stretch* shows elastic deformation. Finally, *wave* presents a real-world waving cloth.

In the following section, we will introduce the evaluation methods that are performed on this dataset, along with the public website[1] to openly evaluate algorithms for the community.

6.3.2 Evaluation Methods and Statistics

Similar to *Middlebury*, we provide tests of *Endpoint Error* (EE) and *Angle Error* (AE). Users are expected to compute flow fields for all frames in the *Long Sequences* group, and calculate one image pair for each sequence in the *Short Sequences* group. For robustness statistics, we perform *Average* (Avg.), *Accumulated* (Acc.), *Standard Deviations* (SD), *RX* and *AX* [5] where Avg., SD and {A50, A75, A99, A100} are given for

[1]The evaluation website will be released to public in April 2014.

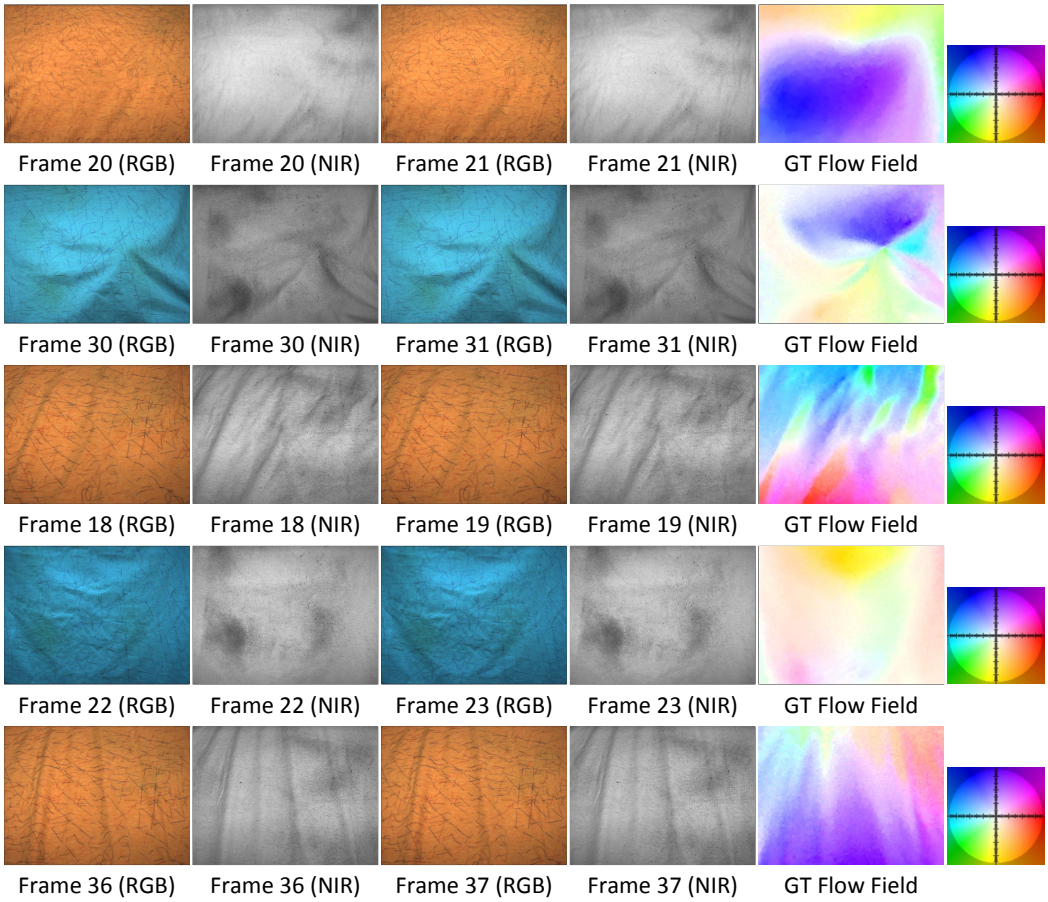


Figure 6-4: Sample frames from the long sequences in our GT dataset. **Top To Bottom**: *mBlur*, *circle*, *crush*, *wave* and *stretch*.

both EE and AE; {R0.5, R0.75, R1, R2} are performed for EE; Acc. is calculated for EE in long sequences only; {R2, R5, R7.5, R10} are computed for AE. Note that the *Accumulated Endpoint Error* (Acc.EE) is the first time proposed in the dense ground truth benchmarks in order to present the algorithm performance against the famous *Drift* issue in long image sequence tracking. The computation of Acc.EE on the k-th frame is formulated as follows:

$$Acc.EE(k) = \sum_{i=1}^{k} \sum_{\mathbf{x}} \frac{\sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}}{n} \tag{6.1}$$

where $\mathbf{w}_i = (u_i, v_i)^T$ and $\hat{\mathbf{w}} = (\hat{u}, \hat{v})^T$ denotes the baseline flow field and ground truth flow field respectively on the i-th frame while n presents the number of ground truth vectors in $\hat{\mathbf{w}}_i$. As shown in Fig. 6-6(a), we generate a comparison table for cross-evaluation against any other methods available on our evaluation system. For long sequences, we can plot results selected by the user with respect to a specific frames index.

6.4 RGB-NIR Variational Optical Flow Model

We now present an optical flow approach which combines RGB-NIR information in such a way as to maximize the distinguishing information from each channel. Certain visual information can be poorly represented in an RGB channel. It is therefore prudent in these cases to consider the NIR channel (and vice-versa). In our experiments section we examine these properties in more detail.

Our algorithm considers a pair of consecutive frames in an image sequence. The current frame is denoted by $I_1(\mathbf{x})$ and its successor is $I_2(\mathbf{x})$ where $I = (V, N)^T$, $\{V : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}\}$ represents a rectangular image in the RGB channel and $\{N : \Omega \subset \mathbb{R}\}$ denotes a rectangular image in the NIR channel. Both V and N are aligned and share the same Cartesian coordinate where $\mathbf{x} = (x, y)^T$ is a pixel location. The optical flow displacement between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ is defined as $\mathbf{w} = (u, v)^T$. Our proposed optical flow approach leads to the following energy function:

$$E(\mathbf{w}) = (1 - \lambda(\mathbf{x}))E_V(\mathbf{w}) + \lambda(\mathbf{x})E_N(\mathbf{w}) + \gamma E_S(\mathbf{w}) \tag{6.2}$$

where the *Visible RGB Energy* $E_V(\mathbf{w})$ contains both *Intensity Constancy* and *Gradient Constancy* assumptions between the visible components $V_1(\mathbf{x})$ and $V_2(\mathbf{x})$ of the images while our main contribution i.e. *Invisible NIR Energy* is represented as the term $E_N(\mathbf{w})$. A high-order regularization $E_S(\mathbf{w})$ is also adopted.

Visible RGB Energy. Following the optical flow assumption regarding *Intensity Constancy*, we assume that the intensity of a pixel is not varied by its displacement throughout an image sequence. In addition, we also make a *Gradient Constancy* assumption [20] to provide additional stability where pixel intensity is violated by illumination changes. The *Visible RGB Energy* term encoding these assumptions is therefore formulated as:

$$\begin{aligned} E_V(\mathbf{w}) = \int_\Omega \phi(&\|V_2(\mathbf{x}+\mathbf{w}) - V_1(\mathbf{x})\|^2 \\ &+ \theta \|\nabla V_2(\mathbf{x}+\mathbf{w}) - \nabla V_1(\mathbf{x})\|^2) d\mathbf{x} \end{aligned} \tag{6.3}$$

For robustness against occlusions and boundary blur, we apply the increasing concave function Charbonnier $\phi(s^2) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$ to solve this formation. The remaining term $\nabla = (\partial_{xx}, \partial_{yy})^T$ is a spatial gradient and $\theta \in [0, 1]$ denotes a linear weight. The smoothness term is a dense pixel based regularizer that penalizes global variation. The objective is to produce a globally smooth constraint:

$$E_S(\mathbf{w}) = \int_\Omega \phi(\|\nabla u\|^2 + \|\nabla v\|^2) d\mathbf{x} \tag{6.4}$$

where the Charbonnier penalty function is employed again. Because Charbonnier penalty is reported capable to give additional robustness on the smooth motion. This penalty is also convex, which yields fast converge speed in the energy minimisation. More analysis on the regularisation can be found in Sec 2.1.3 and Tab. 2-2(a).

Invisible NIR Energy. A *visible RGB Energy* term is widely used in optical flow [21] but error-prone in featureless regions or unclear boundaries. We therefore propose to inspect additional spectral channels given these situations. We include an *Invisible NIR Energy* term as a complementary assumption to the classic framework, namely to introduce additional texture information for the optical flow estimation. Similar to the RGB *Intensity Constancy*, we assume that the intensity of the pixel in the NIR channel is not changed by displacement, which leads to an energy term shown as follows:

$$E_N(\mathbf{w}) = \int_\Omega \phi(\|N_2(\mathbf{x}+\mathbf{w}) - N_1(\mathbf{x})\|^2) d\mathbf{x} \tag{6.5}$$

Where the term $E_N(\mathbf{w})$ presents the continuous energy in the NIR channel. Note that both terms $E_V(\mathbf{w})$ and $E_N(\mathbf{w})$ share the same spatial smoothness regularizer.

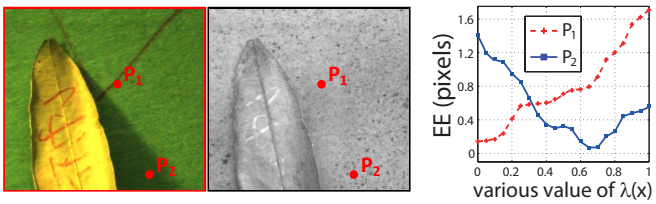


Figure 6-5: *Endpoint Error* (EE) affected by varying weight $\lambda(\mathbf{x})$. **(a)** and **(b)**: A patch of *LeafShadow* is shown where two points of P_1 and P_2 are plotted in RGB and NIR channels respectively. **(c)** EE for both points P_1 and P_2 are plotted by varying weight $\lambda(\mathbf{x})$.

Detail-Aware Weight $\lambda(\mathbf{x})$. In Fig. 6-5(a,b) we show an image patch in which two points P_1 and P_2 are plotted. The small region centred on P_2 contains soft shadow in the RGB channel but has more distinguishing features in the NIR channel. For the point P_1, the situation is opposite. The *Endpoint Error* (EE) with respect to the different $\lambda(\mathbf{x})$ values are plotted in Fig. 6-5(c). We observe that featureless texture leads to a larger error in the optical flow computation. Dynamically taking more contribution from the channel containing more detailed texture is therefore adopted to improve this issue.

6.4.1 minimisation Framework

Prior to energy minimisation, $\lambda(\boldsymbol{x})$ *Initialization* is performed to improve overall optical flow energy in featureless regions. A numerical scheme is then applied to minimise the continuous RGB-NIR energy within a pyramidal framework. In the following sections, both steps are described in detail.

$\lambda(\mathbf{x})$ Initialization. Inspired by the kernel-based edge detector where an *Intensity Gradient* is used to represent geometric information in the texture space, we define a weight $\{\lambda(\mathbf{x}) : \mathbb{R} \mapsto [0,1]\}$ using an *Intensity Gradient* as follows:

$$\lambda(\mathbf{x}) = \left(1 + \exp\left\{-a\left(\frac{|\Delta N_1(\mathbf{x})|}{|\Delta V_1(\mathbf{x})| + |\Delta N_1(\mathbf{x})|} - b\right)\right\}\right)^{-1}$$

where $\mathbf{x}$ denotes a pixel location while $\Delta = (\Delta_x, \Delta_y)^T$ presents the intensity gradient calculated using a 3×3 *Sobel Kernel*; a and b are parameters chosen to be 10 and 0.5 respectively. The weight $\lambda(\mathbf{x})$ is intensity-dependent and can be precalculated before energy minimisation. Given an n-level image pyramid, the input images I_1, I_2 and the weight map $\lambda(\mathbf{x})$ are resized to the same scale on each level. These are denoted by $I_1^i = (V_1^i, N_1^i)^T$, $I_2^i = (V_2^i, N_2^i)^T$ and λ^i, and are used in the following energy minimisation phase.

RGB-NIR energy optimisation. In this process, we aim to find the global minimum of the energy in Eq. (6.2) which is continuous but highly nonlinear. The minimisation scheme for such energy is well studied in the vision community. After *Euler-Lagrange Equations* are employed, we apply nested fixed point iterations on $\mathbf{w}$ by mainly following the numerical scheme in [20]. We define the mathematical abbreviations on both V and N as follows:

$$\begin{aligned}
V_x &= \partial_x V_2(\mathbf{x}+\mathbf{w}) & V_{yy} &= \partial_{yy} V_2(\mathbf{x}+\mathbf{w}) \\
V_y &= \partial_y V_2(\mathbf{x}+\mathbf{w}) & V_z &= V_2(\mathbf{x}+\mathbf{w}) - V_1(\mathbf{x}) \\
V_{xx} &= \partial_{xx} V_2(\mathbf{x}+\mathbf{w}) & V_{xz} &= \partial_x V_2(\mathbf{x}+\mathbf{w}) - \partial_x V_1(\mathbf{x}) \\
V_{xy} &= \partial_{xy} V_2(\mathbf{x}+\mathbf{w}) & V_{yz} &= \partial_y V_2(\mathbf{x}+\mathbf{w}) - \partial_y V_1(\mathbf{x}) \\
N_x &= \partial_x N_2(\mathbf{x}+\mathbf{w}) & & \\
N_y &= \partial_y N_2(\mathbf{x}+\mathbf{w}) & N_z &= N_2(\mathbf{x}+\mathbf{w}) - N_1(\mathbf{x})
\end{aligned}$$

We minimise the optical flow energy $E(\mathbf{w})$ in a coarse-to-fine framework within a top-down image pyramid. In the outer fixed point iterations, the flow field is initialized as $\mathbf{w} = (0,0)^T$ on the top (coarsest) level of the pyramid and updates $\mathbf{w}^{i+1} = \mathbf{w}^i + d\mathbf{w}^i$

to the next finer level. We then apply first order Taylor expansion on the terms V_*^{i+1} and N_*^{i+1}, which results in $V_z^{i+1} \approx V_z^i + V_x^i du^i + V_y^i dv^i$ and $N_z^{i+1} \approx N_z^i + N_x^i du^i + N_y^i dv^i$ where du^i and dv^i are two unknown increments. Inner fixed point iterations are then performed to solve these unknowns. Given the initialization of $du^{i,0} = 0$ and $dv^{i,0} = 0$, we assume that $du^{i,j}$ and $dv^{i,j}$ converge within j iterations. We have the final linear system in $du^{i,j+1}$ and $dv^{i,j+1}$ as follows:

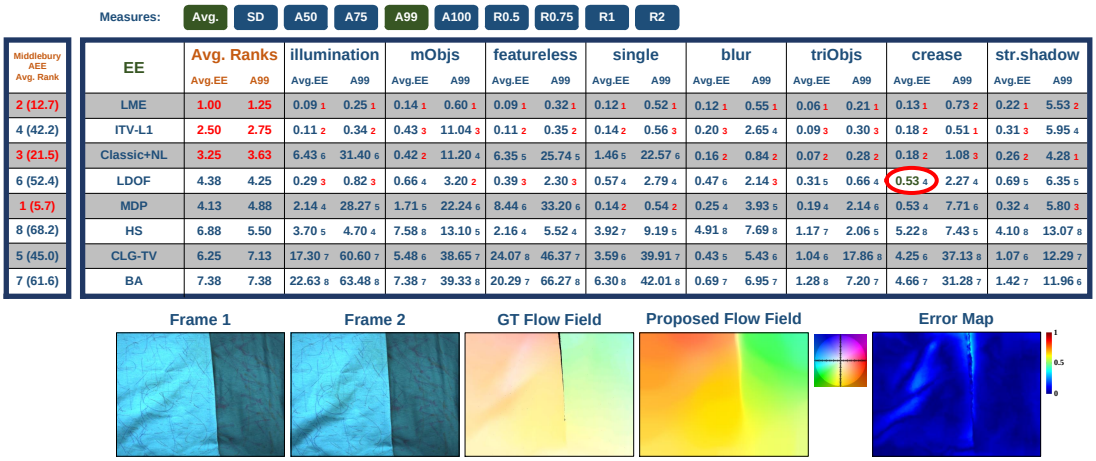
$$\begin{aligned}(1-\lambda^i)(\phi')_V^{i,j} \cdot (V_x^i(V_z^i + V_x^i du^{i,j+1} + V_y^i dv^{i,j+1}) \\ +\theta\ [V_{xx}^i(V_{xz}^i + V_{xx}^i du^{i,j+1} + V_{xy}^i dv^{i,j+1}) \\ +V_{xy}^i(V_{yz}^i + V_{xy}^i du^{i,j+1} + V_{yy}^i dv^{i,j+1})]) \\ +\lambda^i(\phi')_N^{i,j} \cdot N_x^i(N_z^i + N_x^i du^{i,j+1} + N_y^i dv^{i,j+1}) \\ -\gamma\ (\phi')_S^{i,j} \cdot \nabla(u^i + du^{i,j+1}) = 0\end{aligned} \tag{6.6}$$

$$\begin{aligned}(1-\lambda^i)(\phi')_V^{i,j} \cdot (V_y^i(V_z^i + V_x^i du^{i,j+1} + V_y^i dv^{i,j+1}) \\ +\theta\ [V_{yy}^i(V_{yz}^i + V_{xy}^i du^{i,j+1} + V_{yy}^i dv^{i,j+1}) \\ +V_{xy}^i(V_{xz}^i + V_{xx}^i du^{i,j+1} + V_{xy}^i dv^{i,j+1})]) \\ +\lambda^i(\phi')_N^{i,j} \cdot N_y^i(N_z^i + N_x^i du^{i,j+1} + N_y^i dv^{i,j+1}) \\ -\gamma\ (\phi')_S^{i,j} \cdot \nabla(v^i + dv^{i,j+1}) = 0\end{aligned} \tag{6.7}$$

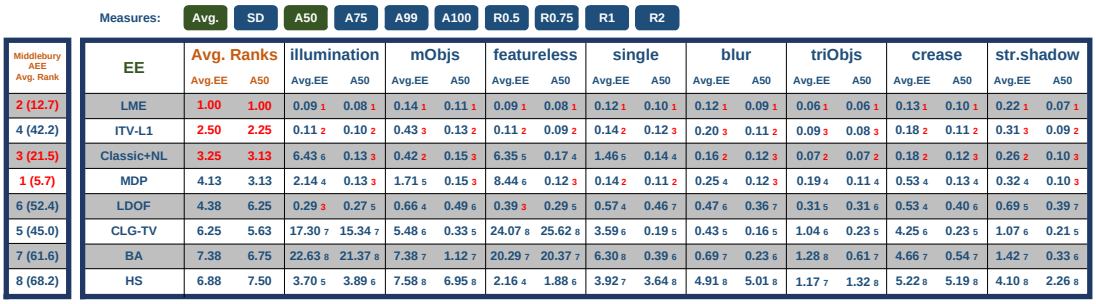
where $(\phi')_V^{i,j}$ and $(\phi')_N^{i,j}$ are interpreted as robustness factors against geometric blur and occlusion on the object boundaries. $(\phi')_S^{i,j}$ represents the diffusivity in the smoothness constraint.

$$\begin{aligned}(\phi')_V^{i,j} &= \phi'((V_z^i + V_x^i du^{i,j} + V_y^{i,j} dv^{i,j})^2 \\ &\quad + \theta[(V_{xz}^i + V_{xx}^i du^{i,j} + V_{xy}^i dv^{i,j})^2 \\ &\quad + (V_{yz}^i + V_{xy}^i du^{i,j} + V_{yy}^i dv^{i,j})^2]) \\ (\phi')_N^{i,j} &= \phi'((N_z^i + N_x^i du^{i,j} + N_y^i dv^{i,j})^2) \\ (\phi')_S^{i,j} &= \phi'(\left\|\nabla(u^i + du^{i,j})\right\|^2 + \left\|\nabla(v^i + dv^{i,j})\right\|^2)\end{aligned}$$

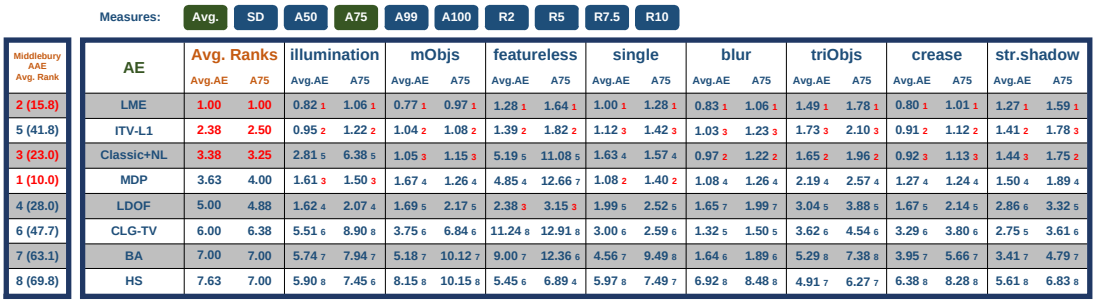
In implementation, the image pyramid is constructed using a downsampling of 0.75. The final linear system Eq. (6.6,6.7) is solved with successive over-relaxation iterations.



(a) Screen shot of our public evaluation website for the short sequences, and illustrating the *Endpoint Error* (EE) evaluation. Multiple statistics/measures (Sec. 6.3.2) can be manually selected on the top of the table and illustrated as sub-columns within a sequence where the subscripts show the rank in that sub-column. The user can mouse-click any of the results to show sequence details, the proposed flow field and the error map against the ground truth. All methods are listed in order of their average rank (Avg. Ranks). Clicking the name of a sequence, methods are re-ranked based on their results for that sequence.



(b) Additional quantitative *Average* (Avg.) and A50 tests in *Endpoint Error* (EE) evaluation.



(c) Additional quantitative *Average* (Avg.) and A75 tests in *Angle Error* (AE) evaluation.

Figure 6-6: Our public evaluation system on the short sequences.

6.5 Experiments

In this section, ***(1)*** we evaluate eight publicly available optical flow algorithms from *Middlebury* using our nonrigid GT dataset, and ***(2)*** our proposed multispectral optical flow method (*vnflow*) is evaluated, highlighting the advantages of using a hybrid RGB-

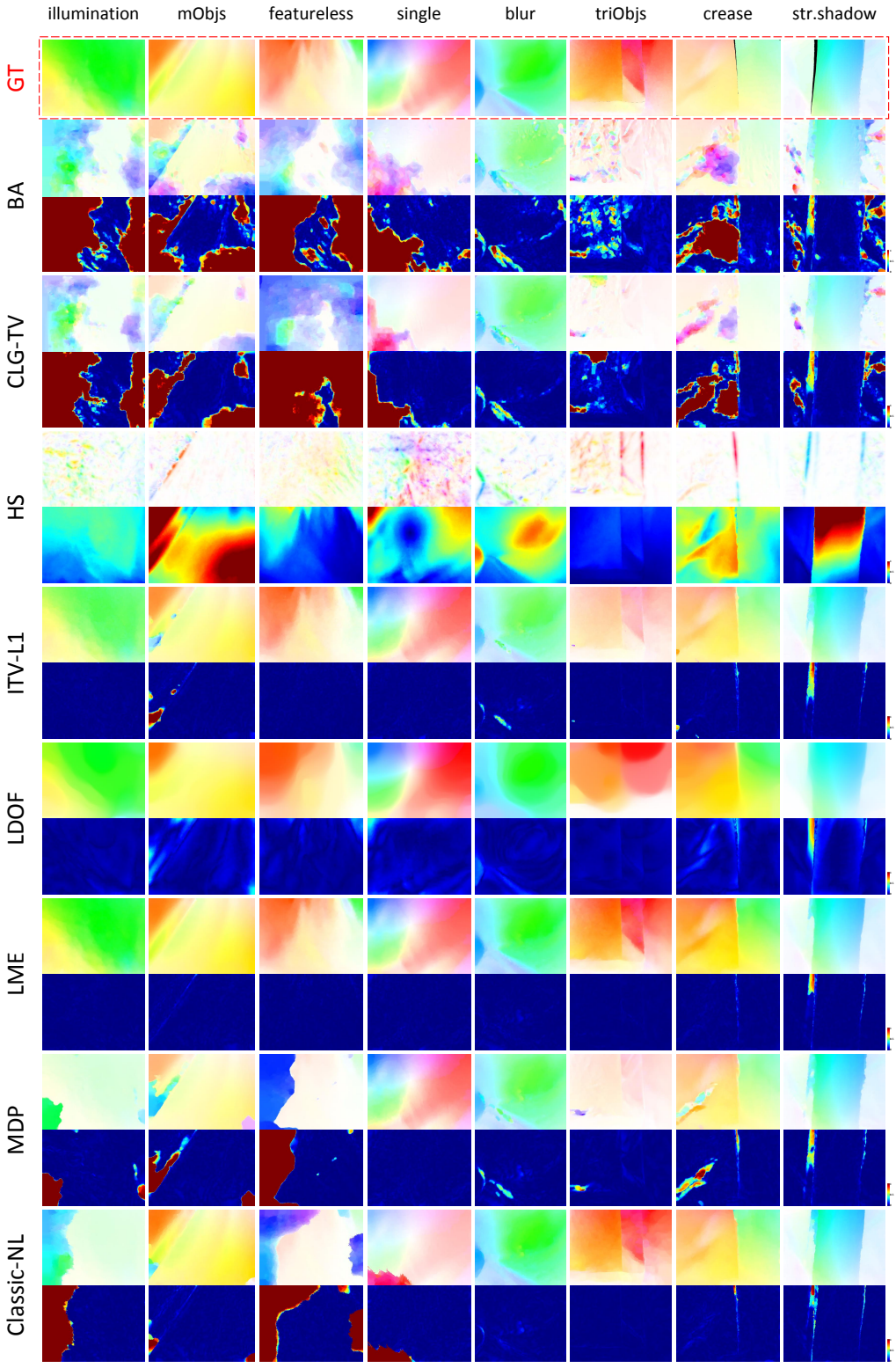


Figure 6-7: Visual comparison of Avg.EE on the short sequences of our ground truth dataset. Both the optical flow fields (**Top**) and the error maps (**Bottom**) are given for each baseline method.

NIR energy scheme. Note that all experiments are performed using a 2.9Ghz Xeon 8-cores, NVIDIA Quadro FX 580, 16Gb memory computer.

We consider eight different baseline methods in our experiments. Algorithms from Xu *et al.* (MDP) [152] (AEE rank 1/81) and our LME approach (rank 6) are the state-of-the-art methods. The former has leading performance in the *Middlebury* evaluation while the latter achieves the state-of-the-art results on Garg *et al.* [49]. *Combined local-global Optical Flow* (CLG-TV) [36] (AIE rank 4/81) highlights the utility of bilateral filtering and anisotropic regularization, which gives high performance in image interpolation. *Large Displacement Optical Flow* (LDOF) [21] (AEE rank 58) is a variational model integrating rich feature descriptors and is designed to overcome large displacement issues. Classic+NL [126] (rank 20) improves the TV-L1 framework by combining a Lorentzian penalty and a median filtering heuristic. *Horn and Schunck* (HS) [55] (rank 75), *Black and Anandan* (BA) [12] (rank 69) and *Improved TV-L1* (ITV-L1) [143] (rank 42) are classic models widely used in real-world image registration.

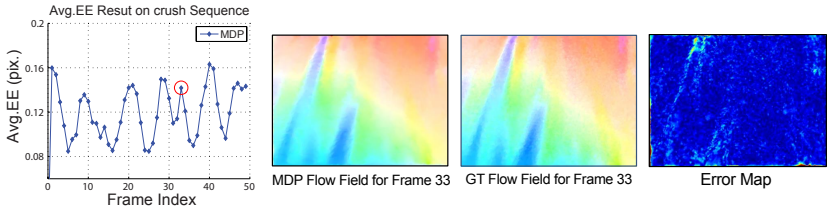
We first perform an evaluation on the short sequences of our GT dataset. Fig. 6-6 shows a screen shot of our public evaluation website where eight optical flow methods are quantitatively compared to each other using their default parameter settings. Note that the relative *Middlebury* AEE rank (Average rank, captured on March 26, 2013) of the baseline methods is also listed for comparison. We observe that LME leads all trials in Avg.EE. ITV-L1 and Classic-NL respectively rank 2.50 and 3.25 in general Avg.EE. The former outperforms most other algorithms in *featureless* while the latter shows more robust toward flow discontinuities (*mObjs*, *triObjs* and *crease*) and blur motion (*blur*). Note that most methods have a large error (>0.5 Avg.EE.) for *illumination* because the strong illumination change violates the *Intensity Consistency*. In this case, LME (Avg.EE 0.09), ITV-L1 (Avg.EE 0.11) and LDOF (Avg.EE 0.29) give higher performance over the other methods, which is visually observed in the comparison shown in Fig. 6-7.

Compared to *Middlebury*, the short sequences of our dataset result in significantly different ranking. We believe this is due to the range of new photometric effects in our GT which are absent in *Middlebury*. MDP achieves the top performance in *Middlebury* but ranks (in relative terms) 6 in *featureless* and 4.13 in Avg.EE by average. This is because large textureless regions in *featureless* provide less SIFT features, in turn weakening the inner motion detail preserving process. In addition, LME ranks higher (in relative terms) than in *Middllebury*. The reason may be due to the local smoothness and the nonrigid deformation penalties, which are preparedly robust to complex nonrigid motion (Avg.EE 0.12 in *blur*) and textureless regions (Avg.EE 0.09 in *featureless*).

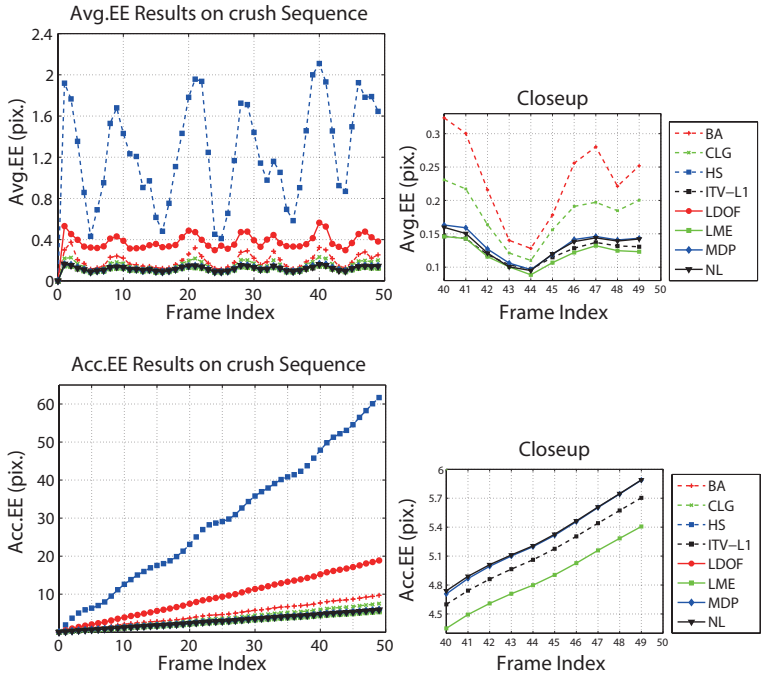
An evaluation on the long sequences is also performed as shown in Fig. 6-8, Fig. 6-9

Table | Graph | Acc. | Avg. | SD | A50 | A75 | A99 | A100 | R0.5 | R0.75 | R1 | R2

EE		Avg. Ranks		mBlur		circle		crush		stretch		wave	
		Avg.EE	SD	Avg.EE	SD	Avg.EE	SD	Avg.EE	SD	Avg.EE	SD	Avg.EE	SD
☑	LME	1.00	1.80	0.16 1	0.33 1	0.13 1	0.53 1	0.11 1	0.13 3	0.09 1	0.09 3	0.09 1	0.07 1
☑	Classic+NL	2.00	2.40	0.23 3	0.52 4	0.14 2	0.56 2	0.12 2	0.12 2	0.09 1	0.08 1	0.11 2	0.21 3
☑	ITV-L1	2.20	2.60	0.20 2	0.34 3	0.18 3	0.97 4	0.12 2	0.11 1	0.09 1	0.08 1	0.12 3	0.56 5
☑	MDP	3.00	4.80	0.25 4	0.60 5	0.21 4	1.05 5	0.12 2	0.16 4	0.09 1	0.10 4	0.19 4	1.27 6
☑	LDOF	5.80	3.60	0.32 5	0.33 1	0.39 5	0.62 3	0.38 7	0.27 6	0.31 7	0.17 6	0.31 5	0.14 2
☑	CLG-TV	5.80	6.20	2.00 6	7.01 7	0.56 6	2.24 7	0.15 5	0.23 5	0.10 5	0.13 5	1.86 7	6.42 7
☑	HS	7.60	6.40	4.13 8	2.09 6	0.93 8	1.07 6	1.26 8	0.81 8	0.56 8	0.39 8	0.77 6	0.54 4
☑	BA	6.80	7.40	3.35 7	8.04 8	0.82 7	2.69 8	0.20 6	0.38 7	0.13 6	0.17 6	3.10 8	7.77 8



(a) **Table View** shows quantitative evaluation on all long sequences. The user can mouse-click any result to bring up the details (**Bottom Row**), in which they are plotted w.r.t. the frame index. Any node within the graph can be clicked to show the visual comparison (ground truth, the proposed flow field and error map) for a specific frame index.



(b) **Graph View** plots details for each sequence. The user can select multiple baseline methods by clicking their checkboxes then clicking the *Graph* option on top of the table. The measure details e.g. Avg.EE and Acc.EE are plotted onto the downloadable graphs for each sequence.

Figure 6-8: Screen shot of our public evaluation website for long sequences, illustrating the *Endpoint Error* (EE) evaluation.

EE	Avg. Ranks		mBlur		circle		crush		stretch		wave	
	Avg.EE	A99	Avg.EE	A99	Avg.EE	A99	Avg.EE	A99	Avg.EE	A99	Avg.EE	A99
☑ LME	1.00	2.00	0.16 1	1.07 1	0.13 1	0.51 1	0.11 1	0.60 3	0.09 1	0.42 3	0.09 1	0.34 2
☑ ITV-L1	2.20	1.60	0.20 2	1.17 2	0.18 3	0.61 3	0.12 2	0.46 1	0.09 1	0.32 1	0.12 3	0.31 1
☑ Classic+NL	2.00	2.60	0.23 3	1.63 4	0.14 2	0.60 2	0.12 2	0.57 2	0.09 1	0.39 2	0.11 2	0.37 3
☑ MDP	3.00	4.60	0.25 4	3.17 5	0.21 4	1.87 5	0.12 2	0.73 4	0.09 1	0.52 4	0.19 4	0.94 5
☑ LDOF	5.80	4.80	0.32 5	1.23 3	0.39 5	1.56 4	0.38 7	1.47 6	0.31 7	0.84 7	0.31 5	0.71 4
☑ CLG-TV	5.80	6.20	2.00 6	31.81 7	0.56 6	12.64 7	0.15 5	1.07 5	0.10 5	0.63 5	1.86 7	37.04 7
☑ BA	6.80	7.40	3.35 7	38.61 8	0.82 7	14.92 8	0.20 6	1.58 7	0.13 6	0.66 6	3.10 8	40.23 8
☑ HS	7.60	6.80	4.13 8	9.32 6	0.93 8	5.18 6	1.26 8	3.64 8	0.56 8	1.99 8	0.77 6	2.59 6

Acc.EE	Avg. Ranks		mBlur		circle		crush		stretch		wave	
	20s Frs	50s Frs	20s Frs	50s Frs	20s Frs	50s Frs	20s Frs	50s Frs	20s Frs	50s Frs	20s Frs	50s Frs
☑ LME	1.00	1.00	3.40 1	7.99 1	1.86 1	6.23 1	2.01 1	5.41 1	1.75 1	4.26 1	1.83 1	4.63 1
☑ Classic+NL	2.60	2.60	4.84 3	11.15 3	2.03 2	7.00 2	2.19 4	5.89 3	1.83 2	4.54 3	2.19 2	5.19 2
☑ ITV-L1	2.60	2.80	4.14 2	9.85 2	2.04 3	9.01 3	2.14 2	5.70 2	1.85 3	4.63 4	2.56 3	5.80 3
☑ MDP	3.80	3.40	5.21 4	12.04 4	2.49 4	10.24 4	2.16 3	5.89 3	1.86 4	4.53 2	4.91 4	9.40 4
☑ CLG-TV	5.60	5.80	42.18 6	98.21 6	5.54 5	27.45 6	2.71 5	7.53 5	2.07 5	4.97 5	46.68 7	92.33 7
☑ LDOF	6.00	5.60	6.18 5	15.56 5	6.32 6	18.99 5	7.01 7	18.86 7	6.29 7	15.16 7	5.91 5	15.40 4
☑ BA	6.80	6.80	68.76 7	164.07 7	9.00 7	40.06 7	3.45 6	9.69 6	2.58 6	6.18 6	75.63 8	152.13 8
☑ HS	7.60	7.60	87.66 8	202.41 8	12.58 8	45.82 8	21.32 8	61.70 8	11.81 8	27.48 8	16.50 6	37.77 6

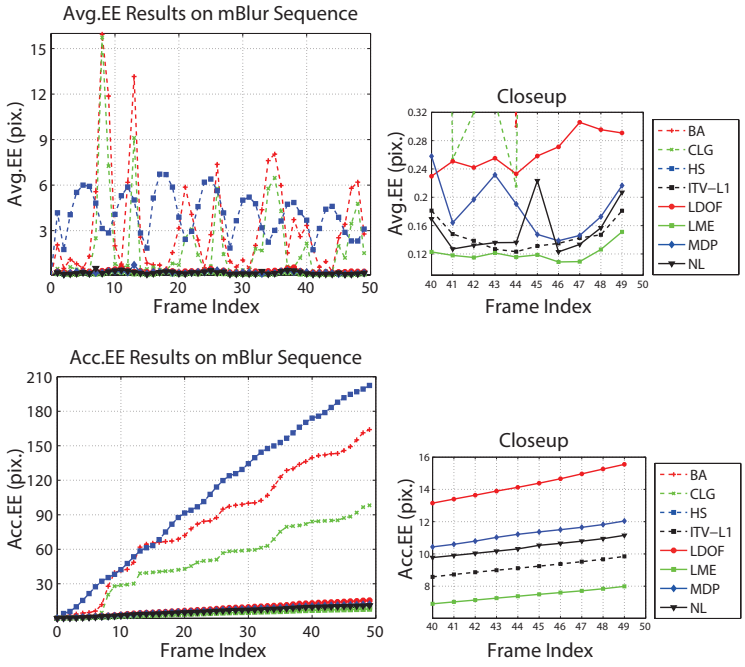


Figure 6-9: Additional *Endpoint Error* (EE) evaluation on the long sequences of our ground truth dataset. **First Row** shows the quantitative evaluation Avg.EE and A99 across all eight baseline methods. **Second Row** illustrates the Acc.EE on the 20th frame and 50th frame respectively. **The Rest** presents the graph view of Avg.EE or Acc.EE plotted details respect to frame index for each sequence. More results can be found in Fig.6-14 and 6-15 in the end of this chapter.

and Fig. 6-10. Similar to the short sequence case, LME provides the best Avg.EE in all trials while Classic+NL, ITV-L1 and MDP yield equally top performance in *stretch*.

AE	Avg. Ranks		mBlur		circle		crush		stretch		wave	
	Avg.AE	R2	Avg.AE	R2	Avg.AE	R2	Avg.AE	R2	Avg.AE	R2	Avg.AE	R2
☑ LME	1.00	1.00	1.03 1	0.07 1	2.46 1	0.50 1	1.85 1	0.34 1	2.70 1	0.56 1	2.34 1	0.49 1
☑ Classic+NL	2.80	2.40	1.14 3	0.09 2	2.59 2	0.53 2	1.95 4	0.38 3	2.76 2	0.59 3	2.38 3	0.52 2
☑ ITV-L1	2.80	3.00	1.13 2	0.10 3	2.59 2	0.53 2	1.94 3	0.38 3	2.85 5	0.61 5	2.37 2	0.52 2
☑ MDP	3.40	3.00	1.19 4	0.10 3	2.69 4	0.54 4	1.91 2	0.36 2	2.77 3	0.57 2	2.64 4	0.55 4
☑ CLG-TV	5.00	4.80	2.16 6	0.26 6	3.13 5	0.61 5	2.10 5	0.42 5	2.78 4	0.59 3	3.63 5	0.66 5
☑ BA	6.60	6.00	3.04 7	0.44 7	3.80 6	0.71 6	2.30 6	0.47 6	3.23 6	0.67 5	4.93 8	0.80 6
☑ LDOF	6.40	6.60	1.53 5	0.23 5	4.21 7	0.82 7	3.53 7	0.80 7	3.93 7	0.85 7	3.77 6	0.81 7
☑ HS	7.80	8.00	5.55 8	0.92 8	4.45 8	0.88 8	4.52 8	0.90 8	4.45 8	0.89 8	4.66 7	0.89 8

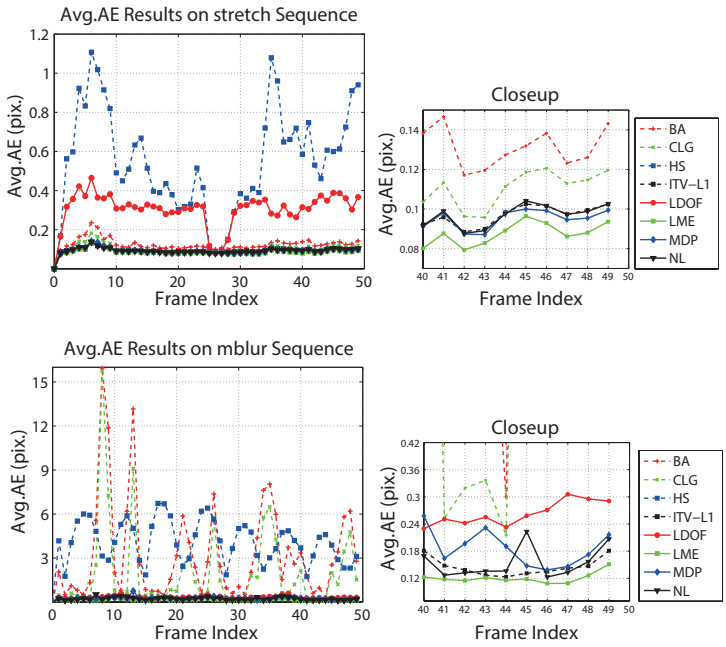


Figure 6-10: Quantitative comparison of *Angle Error* (AE) on the long sequences of our ground truth dataset. Both Table View (**Top Table**) and the Graph View (**The Rest**) are given for each baseline method. More results can be found in Fig.6-16 in the end of this chapter.

EE	Avg. Ranks		illumination		mObjs		featureless		single		blur		triObjs		crease		str.shadow	
	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100	Avg.EE	A100
vnflow.DA	1.00	1.50	0.01 1	0.10 1	0.02 1	0.53 1	0.02 1	0.27 2	0.02 1	0.19 1	0.02 1	0.45 1	0.01 1	0.23 2	0.04 1	8.72 3	0.03 1	6.48 1
vnflow <1>	3.75	3.75	0.54 4	7.24 4	1.59 4	16.43 4	0.12 4	2.05 4	0.10 3	0.93 3	0.16 4	6.10 4	0.15 4	3.64 4	0.39 4	8.75 4	0.22 3	7.38 3
vnflow <0.5>	5.00	4.63	1.07 5	14.43 5	4.14 5	28.04 5	22.19 5	41.02 5	0.18 5	1.83 5	0.28 5	6.18 5	0.29 5	6.70 5	0.65 5	9.21 5	0.35 5	7.29 2
vnflow <0>	6.00	5.75	2.15 6	27.15 6	5.03 6	58.14 6	44.37 6	82.01 6	0.34 6	3.62 6	0.51 6	6.96 6	0.58 6	13.01 6	1.18 6	14.52 6	0.62 6	7.57 4
LME.NIR	2.75	2.13	0.04 2	0.11 2	0.07 2	0.68 2	0.05 2	0.19 1	0.05 2	0.20 2	0.10 2	3.74 3	0.04 2	0.19 1	0.09 2	4.47 1	0.13 2	8.49 5
LME.RGB	3.13	3.25	0.09 3	0.47 3	0.14 3	2.63 3	0.09 3	0.82 3	0.12 4	1.11 4	0.12 3	1.10 2	0.06 3	0.60 3	0.13 3	5.31 2	0.22 3	9.21 6

Figure 6-11: Avg.EE and A100 results of *vnflow* self-comparison: *Detail-Aware Weight* (**DA**) versus the fixed weights (**0**, **0.5** and **1**).

All the methods display comparatively larger Avg.EE in *mBlur* due to the camera blur and fast motion in the scene. In the robustness test (SD), ITV-L1 reaches the top performance on both *crush* and *stretch* while LME yields the best results on the other sequences. Our graph view in Fig. 6-9 shows that LME gives lower accumulated error (Acc.EE) than all other baselines in all the trials while ITV-L1 shows high performance

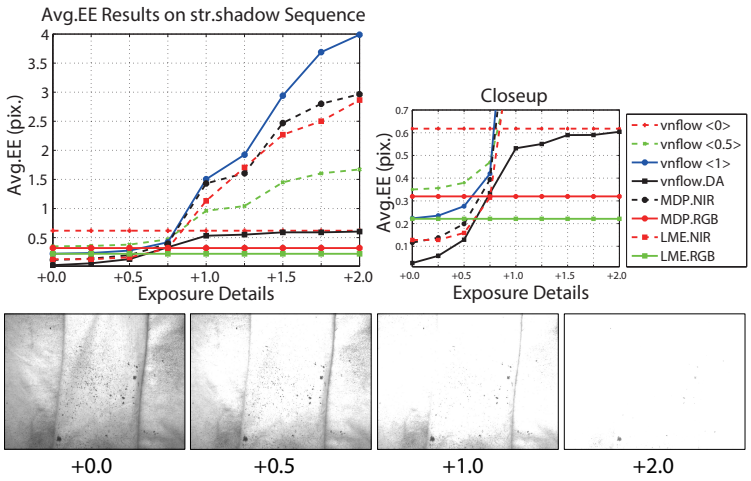


Figure 6-12: Avg.EE measures for *vnflow* on *str.shadow* sequence by varying the exposure (feature distribution) in the NIR channel.

along the entire *crush* sequence.

To evaluate our hybrid RGB-NIR optical flow algorithm – the benefit of using multiple spectrum – we compare our method with the proposed *Detail-Aware Weight* (*vnflow.DA*) against three other implementations using fixed weights (0, 0.5 and 1) in Fig. 6-11. The figure also shows the result of LME applied on the RGB and NIR channels respectively. It is observed that *vnflow.DA* outperforms all other baseline methods in Avg.EE in all cases. Our algorithm **without** NIR energy ($\lambda = 0$) shows low overall performance (Avg.EE rank 6.00) while **with only** NIR energy ($\lambda = 1$) it ranks 3.75 in Avg.EE. In addition, LME with NIR imagery achieves comparably lower overall Avg.EE. In addition, LME with NIR imagery achieves comparably lower overall Avg.EE but shows large A100 error in *str.shadow* due to the large shadow that affects the inner detail preservation process.

We perform an Avg.EE comparison of LME, MDP and four *vnflow* implementations on *str.shadow* by varying the feature distribution in the NIR channel. As shown in Fig. 6-12, we are ramping up the exposure to reduce the overall number of NIR features in the image. As expected, less NIR information (higher exposure) generally increases the Avg.EE. However, even with a very low quantity of NIR information (+2.0), *vnflow.DA* still shows improvement over other implementations using the fixed weights (0, 0.5 and 1).

Finally, Fig. 6-13, a compelling illustration, explains how switching between RGB and NIR information in optical flow can contribute to the strong performance of *vnflow.DA*. Note that those images are captured by our *RGB-NIR Imaging System* with full resolution 1296×966. Our *vnflow.DA* algorithm uses texture details invisible in the RGB channel (second row) where required (and vice-versa). This provides an explanation to why the algorithm performs better against other methods

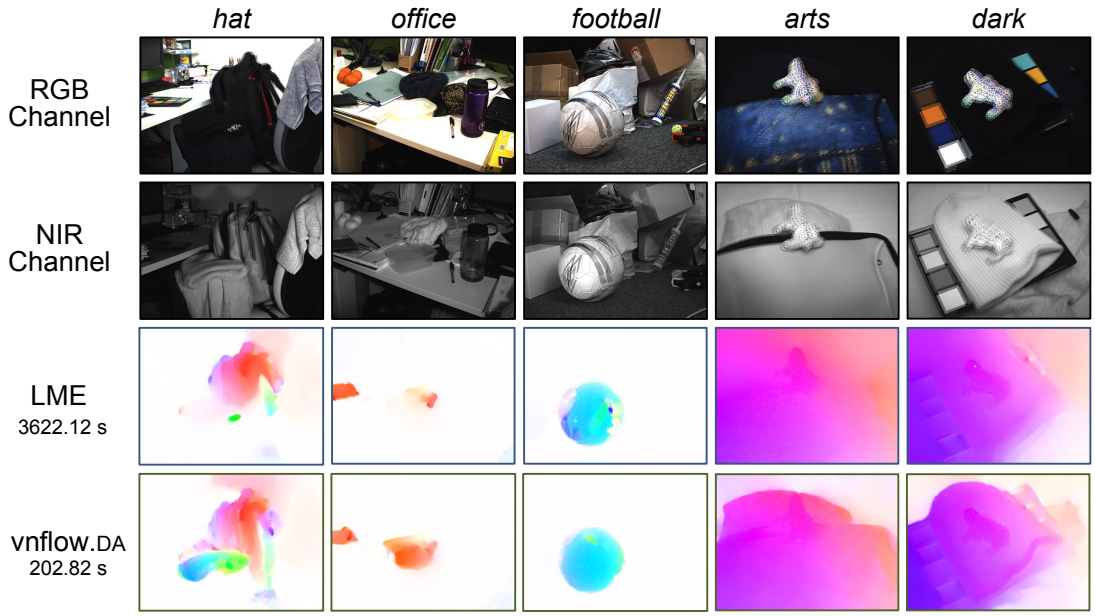


Figure 6-13: Visual comparison of *vnflow.DA* and LME on five real-world sequences of *hat*, *office*, *football*, *arts* and *dark* respectively. Computational time (in second) is given as a number under the names of methods.

which are using either the RGB or NIR channels alone. However, it should be noted that any RGB-NIR evaluation other than the relative one we present would require a *third* hidden spectrum. This may not be practical until multispectral tracking, hardware and other suitable dyes become more widespread in the community. Note that more details of visual comparisons can be seen in the corresponding video footage of http://www.cs.bath.ac.uk/~wl281/vngt/vnGT.mp4.

6.6 Conclusion

In this chapter, we present a new publicly available ground truth dataset for evaluating RGB/Color based optical flow algorithms. By leveraging RGB-NIR imaging and NIR visible dyes, our dataset provides dense ground truth for real-world objects in short and long sequences, as well as with nonrigid motion, illumination changes and motion blur. Algorithms are executed on the RGB sequences, and their result is compared to the ground truth obtained by analysing the dense patterns only visible in the NIR channel. We also propose an optical flow framework which for the first time combines information from ***Multiple*** spectrum in order to optimise overall performance. This provides a compelling insight into the potential benefits for tracking in multiple spectra. One further challenge is finding a dye solution which remains invisible in the RGB channel for any object surface. This way, ground truth deformations could be obtained from a wider range of material.

The related non-refereed report is shown as follows:

W. Li, D. Cosker, and M. Brown, *A Nonrigid Ground Truth Dataset and Multispectral Optical Flow Estimation using Combined RGB and Near-Infrared Imaging*, MTRC Technical Report, University of Bath, March 2013, pp. 1–8.

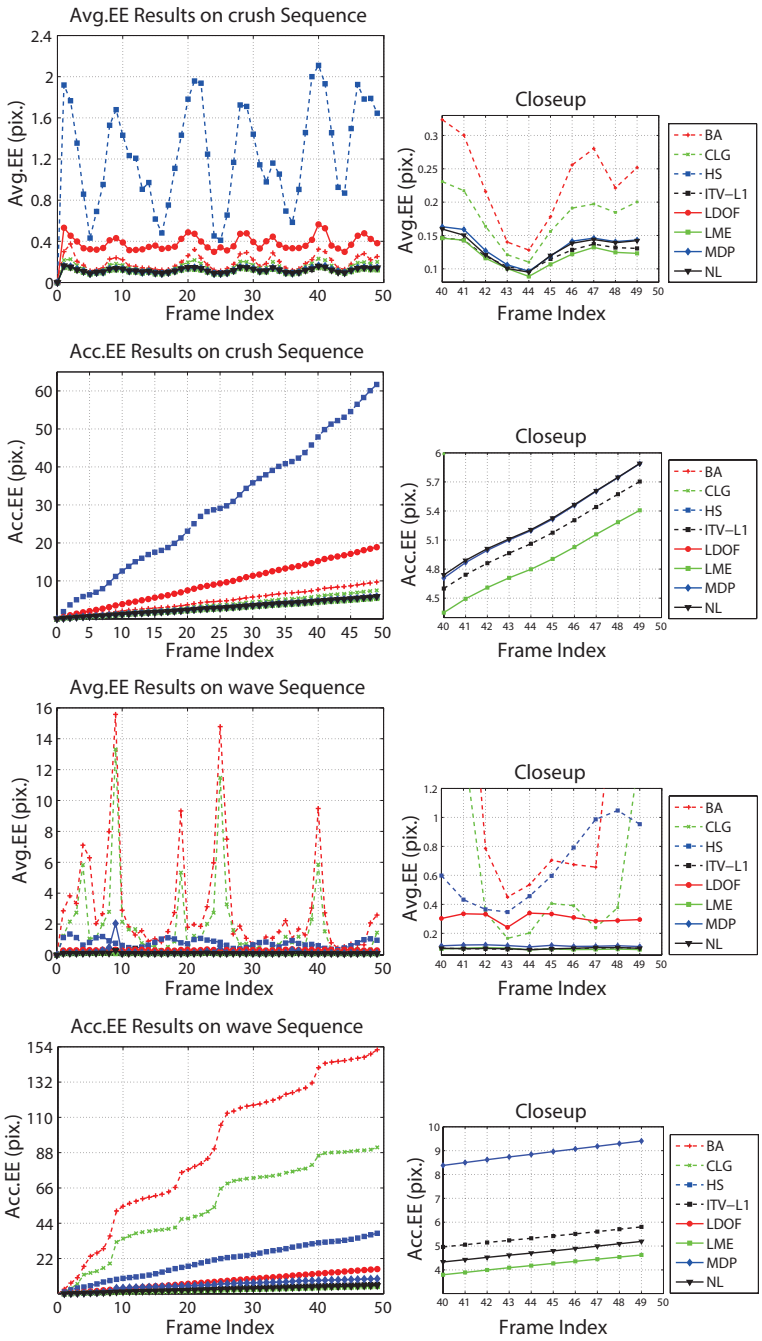


Figure 6-14: Additional results (**Graph View**, plotted details) of Avg.EE and Acc.EE respect to frame index for each sequence.

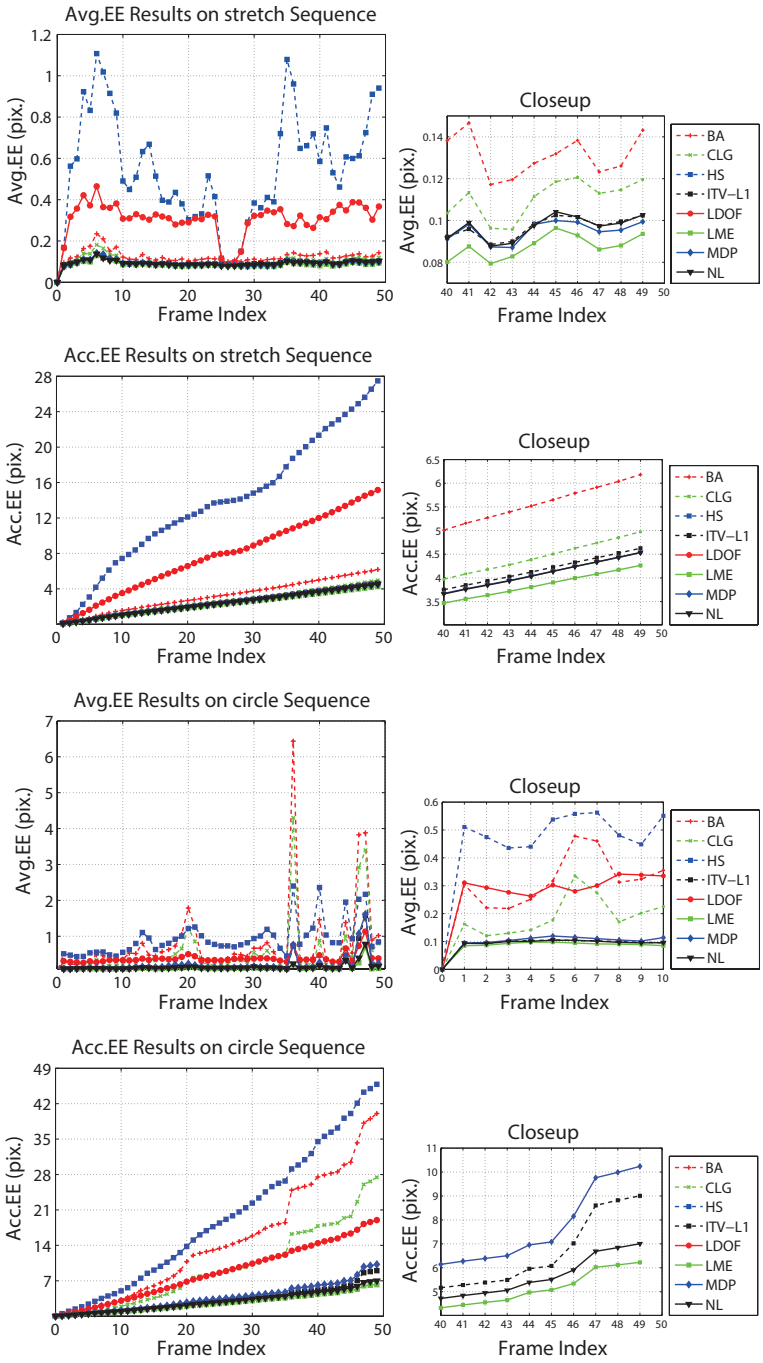


Figure 6-15: Additional results (**Graph view**, plotted details) of Avg.EE and Acc.EE respect to frame index for each sequence.

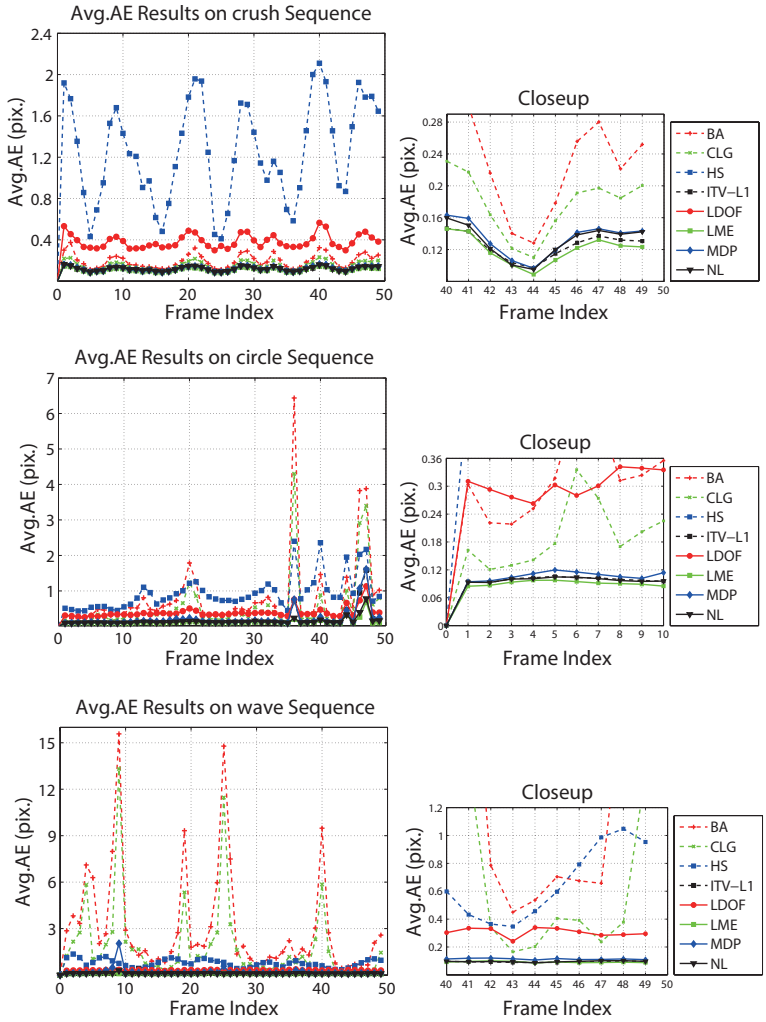


Figure 6-16: Additional results (**Graph view**, plotted details) of Avg.AE respect to frame index for each sequence.

Chapter 7

Conclusions

Dense nonrigid surface tracking in both long and short sequences still poses many challenged research issues and lacks quantitative evaluation. In this work we have studied dense nonrigid surface tracking in both pairwise and long image sequences given scene blur due to camera shake, as well as the potential of quantitative evaluation using the ground truth from hidden features. In this chapter we first summarise the main contributions, which is followed by detailed discussion of the potential further research.

7.1 Main Contributions

One desirable property of this work is to preserve local image details and also handle nonrigid deformations in a long image sequence. A powerful paradigm is interleaving a suitable geometric model and the optical flow energy formula in order to penalise the nonrigid deformation and enhance small motion details. In Chapter 3, we presented a variational optical flow model, together with a novel constraint using *Laplacian Mesh* representation of nonrigid surfaces. Unlike the widely adopted global constraints, our *Laplacian Mesh Constraints* expressed in Laplacian coordinates encourage the local geometric behaviour between the pixel and the adjacent neighbours, and thus preserve the local continuity of optical flow estimated on nonrigid deformations. The approach provides excellent performance on nonrigid surfaces given image boundaries and textureless regions – outperforming or showing comparable accuracy against most the state-of-the-art approaches in several leading publicly available benchmarks.

Real-world nonrigid surface deformation is often accompanied by natural noise which brings additional difficulty when tracking. Camera shake blur is one such example of noise, and often occurs in fast camera movement with low-light conditions due to the requirement of longer exposure times. Even though optical flow models have been extensively studied in the last two decades, existing optical flow approaches find difficult when dealing with blurry scenes because the *Intensity Constancy* assumption

is usually violated. In Chapter 4, we highlight that inter-frame blur in a video is near linear, and may be roughly derived by the 3D camera motion. Such camera motion is captured by our novel *RGB-Motion Imaging System* and adopted as a directional constraint into a blur robust optical flow framework. Our experiments show the high performance of this method against large image blur in both noisy and real-world cases.

Even though we achieve high accuracy of inter-frame tracking on nonrigid surfaces, tracking in long sequences is still difficult because the small error between image pairs is often accumulated over time, which leads to the common *drift* problem. In Chapter 5, we introduce an optimisation framework with automatic *Anchoring* scheme which labels the reliable patches and frames in the entire sequence by interleaving the dense optical flow and the sparse feature matching technique. Our strategy is to shorten tracking distances for local regions, as well as enable parallel tracking throughout a long sequence. Our experiments demonstrate the success in significantly reducing the tracking error of existing optical flow algorithms given synthetic occlusions and image noise in a range of benchmarks.

Dense ground truth for pairwise correspondence is increasingly important for quantitative evaluation of nonrigid surface tracking, particularly given realistic photometric effects with natural image noise. Although current ground truth datasets provide some valuable features, most of them are still limited by the lack of object blur, complex nonrigid deformation and long video sequences. In Chapter 6, we construct dense ground truth for long real-world sequences by simultaneously capturing near-infrared image sequences where the dense markers – visible only under the infrared spectrum – represent the ground truth positions, allowing comparison between the RGB tracked positions and the formation of error metrics. This protocol may also be adopted to capture other types of deformable objects, thus opening ground truth opportunities in other difficult-to-track problems. The capture of real-world objects yields realistic photometric effects - such as blur and illumination change - as well as occlusion and complex deformations. A public evaluation website is constructed to allow for ranking of RGB image based optical flow and other dense tracking algorithms, with varying statistical measures.

7.2 Future Research

This thesis has given several insights and new ideas into nonrigid surface tracking. In this final section, we illustrate the considerable scope for extension of this thesis in three further research directions.

In Chapter 3 effort has been made into incorporating a local spacial geometric constraint using *Laplacian Mesh* representation and processing. This shows the success in nonrigid surface tracking. However, *Laplacian Mesh* is not the only representation in

the area of mesh processing, as well as contains unexpected errors given self occlusions. For the further attention, exploiting more innovative representations, as well as more spatial-temporal constraints e.g. temporal pixel trajectory, may be more successful.

In both Chapters 4 and 6, we demonstrate the potential of multiple sensing imaging to improve optical flow estimation against real-world difficulties i.e. image blur and large textureless regions. The utilisation of an additional sensing channel may offer many opportunities in classic problems, in particular when it is applied to biological perception. The human perception system contains multiple sensing channels which are interleaved to conduct the best environmental perception. Bringing more sophisticated sensing techniques is promising for further research in the general computational perception area.

In Chapter 6 we introduce hidden feature conduction using infrared visible dyes. In this case, the dense feature map brings more tracking information in the invisible infrared spectrum. Accompanying the RGB-NIR multiple sensing imaging system, we construct a dense ground truth for nonrigid surfaces. However, the current dyes used in this thesis can only fit cloth made by cotton or polyester but partially absorb the green band of the visible spectrum on other material surfaces. There are many other dye solutions we could consider – and their related spectrums – where the dense markers would still remain invisible in the RGB channel. Such further investigation may extend our range of ground truth capture.

Appendix A

Derivations of Hybrid Optical Flow Models

In this appendix, we illustrate the detailed derivations of our three hybrid optical flow models involved in this thesis.

A.1 Laplacian Mesh Energy Optimisation

As described in chapter 3, we introduce a novel *Laplacian Mesh Energy* and an improved coarse-to-fine framework for optical flow estimation in both multiple objects and non-rigid cases. Our energy function is shown as follows:

$$E(\mathbf{w}) = E_{Data}(\mathbf{w}) + \lambda E_{Lap}(\mathbf{w}) + \xi E_{Smooth}(\mathbf{w}) \tag{A.1}$$

Where the $E_{Data}(\mathbf{w})$ and $E_{Smooth}(\mathbf{w})$ respectively present the *Continuous Intensity Energy* and high-ordered smoothness while $E_{Lap}(\mathbf{w})$ describes our novel *Laplacian Mesh Energy* that is a discrete term related to the sparse input mesh.

A.1.1 Continuous Laplacian Mesh Energy Estimation

To minimise this hybrid energy, we first represent the vector space of $E_{Lap}(\mathbf{w})$ using polar coordinates denoted by $\mathcal{L} = (\mathcal{L}_r, \mathcal{L}_\theta)^T$ where $\mathcal{L} = (X, Y)^T$, which denotes the vector space of Laplacian coordinates while $\mathcal{L}_r$ denotes the magnitude component and $\mathcal{L}_\theta$ denotes the angle component. We have

$$\mathcal{L}_r = \sqrt{X^2 + Y^2} \tag{A.2}$$

$$\mathcal{L}_\theta = arctan(\frac{Y}{X}) \tag{A.3}$$

which results in two terms for the *Laplacian Mesh Energy* as follows:

$$\begin{aligned} E_{Lap}(\mathbf{w}) = \lambda \int_\Omega \psi(\|\mathcal{L}_{r\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{r\cdot 1}(\mathbf{x})\|^2) d\mathbf{x} \\ + \lambda \int_\Omega \psi(\|\mathcal{L}_{\theta\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{\theta\cdot 1}(\mathbf{x})\|^2) d\mathbf{x} \end{aligned} \tag{A.4}$$

Where the terms $\mathcal{L}_{*\cdot 2}$ and $\mathcal{L}_{*\cdot 1}$ are respectively computed using the meshes $\mathcal{M}_2^k$ and $\mathcal{M}_1^k$. Term $\mathcal{M}_2^k$ is estimated in the *Frame-Frame Tracked Mesh* $\mathcal{M}_2$ *Estimation* step while term $\mathcal{M}_1^k$ is computed in *Edge-Aware Mesh Initialization* step and resized to current level of the image pyramid. Note that the terms $\mathcal{L}_{*\cdot 2}(\mathbf{x}+\mathbf{w})$ and $\mathcal{L}_{*\cdot 1}(\mathbf{x})$ are applied on pixels of the input images using bi-cubic interpolation.

A.1.2 Numerical Scheme for Hybrid Energy optimisation

As mentioned in Chapter 3, we follow an improved coarse-fine-framework and nested fixed point iterations to minimise the hybrid energy. Our numerical scheme is similar to Brox *et al.* [20]. For better description, we refer to the abbreviations from the chapter as follows:

$$\begin{aligned} I_x &= \partial_x I(\mathbf{x}+\mathbf{w}) & I_{yy} &= \partial_{yy} I(\mathbf{x}+\mathbf{w}) \\ I_y &= \partial_y I(\mathbf{x}+\mathbf{w}) & I_{xx} &= \partial_{xx} I(\mathbf{x}+\mathbf{w}) \\ I_z &= I_2(\mathbf{x}+\mathbf{w}) - I_1(\mathbf{x}) & I_{xz} &= \partial_x I_2(\mathbf{x}+\mathbf{w}) - \partial_x I_1(\mathbf{x}) \\ I_{xy} &= \partial_{xy} I(\mathbf{x}+\mathbf{w}) & I_{yz} &= \partial_y I_2(\mathbf{x}+\mathbf{w}) - \partial_y I_1(\mathbf{x}) \end{aligned}$$

$$\begin{aligned} \mathcal{L}_{*\cdot x} &= \partial_x \mathcal{L}_*(\mathbf{x}+\mathbf{w}) \\ \mathcal{L}_{*\cdot y} &= \partial_y \mathcal{L}_*(\mathbf{x}+\mathbf{w}) & \mathcal{L}_{*\cdot z} &= \mathcal{L}_{*\cdot 2}(\mathbf{x}+\mathbf{w}) - \mathcal{L}_{*\cdot 1}(\mathbf{x}) \end{aligned}$$

The first step for minimising the energy is to apply the Euler-Lagrange on the Eq. A.1, we have

$$\begin{aligned}&\psi'(I_z^2+\theta(I_{xz}^2+I_{yz}^2))\cdot(I_xI_z+\theta(I_{xx}I_{xz}+I_{xy}I_{yz}))\\&+\lambda\psi'(\mathcal{L}_{r\cdot z}^2)\cdot(\mathcal{L}_{r\cdot x}\mathcal{L}_{r\cdot z})+\lambda\psi'(\mathcal{L}_{\theta\cdot z}^2)\cdot(\mathcal{L}_{\theta\cdot x}\mathcal{L}_{\theta\cdot z})\\&-\xi\mathbf{Div}(\varphi'(\|\nabla u\|^2+\|\nabla v\|^2)\cdot\nabla u)=0\end{aligned}\tag{A.5}$$

$$\begin{aligned}&\psi'(I_z^2+\theta(I_{xz}^2+I_{yz}^2))\cdot(I_yI_z+\theta(I_{yy}I_{yz}+I_{xy}I_{xz}))\\&+\lambda\psi'(\mathcal{L}_{r\cdot z}^2)\cdot(\mathcal{L}_{r\cdot y}\mathcal{L}_{r\cdot z})+\lambda\psi'(\mathcal{L}_{\theta\cdot z}^2)\cdot(\mathcal{L}_{\theta\cdot y}\mathcal{L}_{\theta\cdot z})\\&-\xi\mathbf{Div}(\varphi'(\|\nabla u\|^2+\|\nabla v\|^2)\cdot\nabla v)=0\end{aligned}\tag{A.6}$$

Next, we construct a n-level image pyramid and go through every level from the top coarsest level with an initial flow field $\mathbf{w}^0=(0,0)^T$. We apply the first fixed point iterations on $\mathbf{w}$, the solution $\mathbf{w}^{k+1}$ can then be obtained by solving the system

$$\begin{aligned}&\psi'((I_z^{k+1})^2+\theta((I_{xz}^{k+1})^2+(I_{yz}^{k+1})^2))\cdot(I_x^kI_z^{k+1}+\theta(I_{xx}^kI_{xz}^{k+1}+I_{xy}^kI_{yz}^{k+1}))\\&+\lambda_1\psi'((\mathcal{L}_{r\cdot z}^{k+1})^2)\cdot\mathcal{L}_{r\cdot x}^k\mathcal{L}_{r\cdot z}^{k+1}+\lambda_2\psi'((\mathcal{L}_{\theta\cdot z}^{k+1})^2)\cdot\mathcal{L}_{\theta\cdot x}^k\mathcal{L}_{r\cdot z}^{k+1}\\&+\xi\mathbf{Div}(\varphi'(\left\|\nabla u^{k+1}\right\|^2+\left\|\nabla v^{k+1}\right\|^2)\cdot\nabla u^{k+1})=0\end{aligned}\tag{A.7}$$

$$\begin{aligned}&\psi'((I_z^{k+1})^2+\theta((I_{xz}^{k+1})^2+(I_{yz}^{k+1})^2))\cdot(I_y^kI_z^{k+1}+\theta(I_{yy}^kI_{yz}^{k+1}+I_{xy}^kI_{xz}^{k+1}))\\&+\lambda_1\psi'((\mathcal{L}_{r\cdot z}^{k+1})^2)\cdot\mathcal{L}_{r\cdot y}^k\mathcal{L}_{r\cdot z}^{k+1}+\lambda_2\psi'((\mathcal{L}_{\theta\cdot z}^{k+1})^2)\cdot\mathcal{L}_{\theta\cdot y}^k\mathcal{L}_{r\cdot z}^{k+1}\\&+\xi\mathbf{Div}(\varphi'(\left\|\nabla u^{k+1}\right\|^2+\left\|\nabla v^{k+1}\right\|^2)\cdot\nabla v^{k+1})=0\end{aligned}\tag{A.8}$$

After the first fixed point iterations, the system of Eq. (A.7, A.8) is still difficult to solve due to the nonlinearity on the terms of I_*^{k+1}, $\mathcal{L}_*^{k+1}$ and function ψ'. First order Taylor expansions are employed on both I_*^{k+1} and $\mathcal{L}_*^{k+1}$. We have

$$\begin{aligned}I_z^{k+1}&\approx I_z^k+I_x^kdu^k+I_y^kdv^k\\I_{xz}^{k+1}&\approx I_{xz}^k+I_{xx}^kdu^k+I_{xy}^kdv^k\\I_{yz}^{k+1}&\approx I_{yz}^k+I_{xy}^kdu^k+I_{yy}^kdv^k\\\mathcal{L}_{*\cdot z}^{k+1}&\approx\mathcal{L}_{*\cdot z}^k+\mathcal{L}_{*\cdot x}^{k+1}du^k+\mathcal{L}_{*\cdot y}^kdv^k\end{aligned}$$

Where we assume that the flow field on level $k+1$ can be estimated by the flow field and the incremental from previous level k, denoted as $\mathbf{w}^{k+1}\approx\widehat{\mathbf{w}}^k+\mathbf{dw}^k$. Note that $\widehat{\mathbf{w}}^k$ is the flow field optimised using $\mathbf{w}^k$ and the remaining small flow details (Sec. 3.3.2). We have a new system as follows:

$$
\begin{aligned}
&(\psi')^k_{Data} \cdot (I^k_x(I^k_z + I^k_x du^k + I^k_y dv^k) \\
+\theta\ &[I^k_{xx}(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k) + I^k_{xy}(I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)]) \\
&+\lambda\ (\psi')^k_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot x}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^k + \mathcal{L}^k_{r\cdot y} dv^k) \\
&+\lambda\ (\psi')^k_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot x}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^k + \mathcal{L}^k_{\theta\cdot y} dv^k) \\
&-\xi\ \mathbf{Div}(\varphi')^k_{Smooth} \cdot \nabla(u^k + du^k) = 0
\end{aligned} \tag{A.9}
$$

$$
\begin{aligned}
&(\psi')^k_{Data} \cdot (I^k_y(I^k_z + I^k_x du^k + I^k_y dv^k) \\
+\theta\ &[I^k_{yy}(I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k) + I^k_{xy}(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)]) \\
&+\lambda(\psi')^k_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot y}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^k + \mathcal{L}^k_{r\cdot y} dv^k) \\
&+\lambda(\psi')^k_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot y}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^k + \mathcal{L}^k_{\theta\cdot y} dv^k) \\
&-\xi\ \mathbf{Div}(\varphi')^k_{Smooth} \cdot \nabla(v^k + dv^k) = 0
\end{aligned} \tag{A.10}
$$

Where $(\psi')^k_{Data}$ and $(\psi')^k_{Lap\cdot *}$ provides the robustness against both the occlusion and the flow blur on object boundaries, $(\varphi')^k_{Smooth}$ is defined as diffusivity in the global smoothness terms [20]. All of those terms are detailed as follows:

$$
\begin{aligned}
(\psi')^k_{Data} &= \psi'((I^k_z + I^k_x du^k + I^k_y dv^k)^2 \\
&\quad + \theta[(I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k)^2 + (I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k)^2]) \\
(\psi')^k_{Lap\cdot *} &= \psi'(\mathcal{L}^k_{*\cdot z} + \mathcal{L}^k_{*\cdot x} du^k + \mathcal{L}^k_{*\cdot y} dv^k)^2 \\
(\varphi')^k_{Smooth} &= \varphi'(\left\|\nabla(u^k + du^k)\right\|^2 + \left\|\nabla(v^k + dv^k)\right\|^2)
\end{aligned} \tag{A.11}
$$

Once a fixed $\mathbf{w}^k$ is reached, the system of Eq. (A.10) still has nonlinearity on ψ'. A nested second fixed point iteration is then applied on $\mathbf{dw}^k$ to remove the nonlinearity of the ψ'. We assume that both $du^{k,j}$ and $dv^{k,j}$ converges in j iteration steps with initialization of $du^{k,0} = 0$ and $dv^{k,0} = 0$. Therefore, the final linear system is obtained in $du^{k,j+1}$ and $dv^{k,j+1}$ as follows:

$$
\begin{aligned}
&(\psi')^{k,j}_{Data} \cdot (I^k_x(I^k_z + I^k_x du^{k,j} + I^k_y dv^{k,j+1}) \\
+\theta\ [I^k_{xx}(I^k_{xz} + I^k_{xx} du^{k,j} + I^k_{xy} dv^{k,j+1}) + I^k_{xy}(I^k_{yz} + I^k_{xy} du^{k,j} + I^k_{yy} dv^{k,j+1})]) \\
&+\lambda\ (\psi')^{k,j}_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot x}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^{k,j} + \mathcal{L}^k_{r\cdot y} dv^{k,j+1}) \\
&+\lambda\ (\psi')^{k,j}_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot x}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^{k,j} + \mathcal{L}^k_{\theta\cdot y} dv^{k,j+1}) \\
&-\xi\ \mathbf{Div}(\varphi')^{k,j}_{Smooth} \cdot \nabla(u^k + du^{k,j+1}) = 0 \quad \text{(A.12)}
\end{aligned}
$$

$$
\begin{aligned}
&(\psi')^{k,j}_{Data} \cdot (I^k_y(I^k_z + I^k_x du^{k,j} + I^k_y dv^{k,j+1}) \\
+\theta\ [I^k_{yy}(I^k_{yz} + I^k_{xy} du^{k,j} + I^k_{yy} dv^{k,j+1}) + I^k_{xy}(I^k_{xz} + I^k_{xx} du^{k,j} + I^k_{xy} dv^{k,j+1})]) \\
&+\lambda\ (\psi')^{k,j}_{Lap\cdot r} \cdot \mathcal{L}^k_{r\cdot y}(\mathcal{L}^k_{r\cdot z} + \mathcal{L}^k_{r\cdot x} du^{k,j} + \mathcal{L}^k_{r\cdot y} dv^{k,j+1}) \\
&+\lambda\ (\psi')^{k,j}_{Lap\cdot \theta} \cdot \mathcal{L}^k_{\theta\cdot y}(\mathcal{L}^k_{\theta\cdot z} + \mathcal{L}^k_{\theta\cdot x} du^{k,j} + \mathcal{L}^k_{\theta\cdot y} dv^{k,j+1}) \\
&-\xi\ \mathbf{Div}(\varphi')^{k,j}_{Smooth} \cdot \nabla(v^k + dv^{k,j+1}) = 0 \quad \text{(A.13)}
\end{aligned}
$$

In order to compute *Div* term that refers to $(\varphi')^{k,j}_{Smooth}$, we have to calculate Laplacian operator and the gradient magnitudes of $|\nabla u|$ and $|\nabla v|$ in image space. Laplacian operator is practically approximated numerically based on finite differences in discrete cases. Hence we have $\nabla u = \overline{u} - u$ and $\nabla v = \overline{v} - v$, where $\overline{u}$ and $\overline{v}$ are weighted average of u or v and calculated by the adjacent neighbourhoods around a specific pixel. The methods to determine $|\nabla u|$ and $|\nabla v|$ have been discussed for many years – finite differences in Faisal and Barron's work [37] is applied to our approach. After obtaining Laplacian operator and the gradient magnitudes, the linear system can be solved by using common numerical methods such as *Gauss-Seidel* and Successive Over Relaxation (*SOR*).

A.2 Blur-Robust Optical flow Energy optimisation

In Chapter 4, we introduce a novel *Blur-Robust Energy* for optical flow estimation between camera blur scenes. The main energy function is given as follows:

$$E(\mathbf{w}) = E_B(\mathbf{w}) + \gamma E_S(\mathbf{w}) \tag{A.14}$$

where $E_B(\mathbf{w})$ represents the intensity energy consisting the *Intensity* and *Gradient Constancy* in the blur image space while $E_S(\mathbf{w})$ denotes a high-ordered smoothness regularization. The non-uniform blur between input images leads to violation on the basic optical flow assumption w.r.t. *Intensity Constancy*. Thus we apply the blur kernel from each input image to the other before the energy minimisation. We have:

$$b_1 = k_2 \otimes I_1 \approx k_2 \otimes k_1 \otimes l_1 \tag{A.15}$$
$$b_2 = k_1 \otimes I_2 \approx k_1 \otimes k_2 \otimes l_2 \tag{A.16}$$

where k_1 is the blur kernel from I_1 while the k_2 is from image I_2. In the following subsection, we give the full details of *Blur-Robust Optical Flow Energy* minimisation on image b_1 and b_2.

A.2.1 Numerical Scheme for Energy minimisation

As mentioned in Chapter 4, a coarse-to-fine strategy with nested fixed point iterations are applied to minimise our proposed *Blur-Robust Optical Flow Energy*. This numerical strategy is widely used in the recent state-of-the-art works [20]. Here, the same abbreviations are referred as follows:

$$\begin{aligned}
b_x &= \partial_x b_2(\mathbf{x}+\mathbf{w}) & b_{yy} &= \partial_{yy} b_2(\mathbf{x}+\mathbf{w}) \\
b_y &= \partial_y b_2(\mathbf{x}+\mathbf{w}) & b_z &= b_2(\mathbf{x}+\mathbf{w}) - b_1(\mathbf{x}) \\
b_{xx} &= \partial_{xx} b_2(\mathbf{x}+\mathbf{w}) & b_{xz} &= \partial_x b_2(\mathbf{x}+\mathbf{w}) - \partial_x b_1(\mathbf{x}) \\
b_{xy} &= \partial_{xy} b_2(\mathbf{x}+\mathbf{w}) & b_{yz} &= \partial_y b_2(\mathbf{x}+\mathbf{w}) - \partial_y b_1(\mathbf{x})
\end{aligned}$$

At the first phase of energy minimisation, a system is built based on Eq. A.14 where Euler-Lagrange is employed as follows:

$$\phi'\{b_z^2 + \alpha(b_{xz}^2 + b_{yz}^2)\} \cdot \{b_x b_z + \alpha(b_{xx} b_{xz} + b_{xy} b_{yz})\} - \gamma\phi'(\|\nabla u\|^2 + \|\nabla v\|^2) \cdot \nabla u = 0 \tag{A.17}$$

$$\phi'\{b_z^2 + \alpha(b_{xz}^2 + b_{yz}^2)\} \cdot \{b_y b_z + \alpha(b_{yy} b_{yz} + b_{xy} b_{xz})\} - \gamma\phi'(\|\nabla u\|^2 + \|\nabla v\|^2) \cdot \nabla v = 0 \tag{A.18}$$

An n-level image pyramid is then constructed from the top coarsest level to the bottom finest level. The flow field is initialized as $\mathbf{w}^0 = (0, 0)^T$ on the top level and the outer fixed point iterations are applied on $\mathbf{w}$. We assume that the solution $\mathbf{w}^{i+1}$ converges on the $i + 1$ level. We have:

$$\begin{aligned}\phi'\{(b_z^{i+1})^2 + \alpha(b_{xz}^{i+1})^2 + \alpha(b_{yz}^{i+1})^2\} \cdot \{b_x^i b_z^{i+1} + \alpha(b_{xx}^i b_{xz}^{i+1} + b_{xy}^i b_{yz}^{i+1})\} \\ -\gamma\phi'(\left\|\nabla u^{i+1}\right\|^2 + \left\|\nabla v^{i+1}\right\|^2) \cdot \nabla u^{i+1} = 0\end{aligned} \tag{A.19}$$

$$\begin{aligned}\phi'\{(b_z^{i+1})^2 + \alpha(b_{xz}^{i+1})^2 + \alpha(b_{yz}^{i+1})^2\} \cdot \{b_y^i b_z^{i+1} + \alpha(b_{yy}^i b_{yz}^{i+1} + b_{xy}^i b_{xz}^{i+1})\} \\ -\gamma\phi'(\left\|\nabla u^{i+1}\right\|^2 + \left\|\nabla v^{i+1}\right\|^2) \cdot \nabla v^{i+1} = 0\end{aligned} \tag{A.20}$$

Because of the nonlinearity in terms of ϕ', b_*^{i+1}, the system (Eqs. A.19, A.20) is difficult to solve by linear numerical methods. We apply the first order Taylor expansions to remove these nonlinearity in b_*^{i+1}, which results in:

$$\begin{aligned} b_z^{i+1} &\approx b_z^i + b_x^i du^i + b_y^i dv^i \\ b_{xz}^{i+1} &\approx b_{xz}^k + b_{xx}^i du^i + b_{xy}^i dv^i \\ b_{yz}^{i+1} &\approx b_{yz}^k + b_{xy}^i du^i + b_{yy}^i dv^i \end{aligned}$$

Based on the coarse-to-fine flow assumption of Brox *et al.* [20] w.r.t. $u^{i+1} \approx u^i + du^i$ and $v^{i+1} \approx v^i + dv^i$ where the unknown flow field on the next level $i+1$ can be obtained using the flow field and its incremental from the current level i. The new system can be presented as follows:

$$
\begin{aligned}
&(\phi')^i_B \cdot \{b^i_x(b^i_z + b^i_x du^i + b^i_y dv^i) \\
&+\alpha b^i_{xx}(b^i_{xz} + b^i_{xx} du^i + b^i_{xy} dv^i) + \alpha b^i_{xy}(b^i_{yz} + b^i_{xy} du^i + b^i_{yy} dv^i)\} \\
&-\gamma(\phi')^i_S \cdot \nabla(u^i + du^i) = 0
\end{aligned} \tag{A.21}
$$

$$
\begin{aligned}
&(\phi')^i_B \cdot \{b^i_y(b^i_z + b^i_x du^i + b^i_y dv^i) \\
&+\alpha b^i_{yy}(b^i_{yz} + b^i_{xy} du^i + b^i_{yy} dv^i) + \alpha b^i_{xy}(b^i_{xz} + b^i_{xx} du^i + b^i_{xy} dv^i)\} \\
&-\gamma(\phi')^i_S \cdot \nabla(v^i + dv^i) = 0
\end{aligned} \tag{A.22}
$$

where the terms $(\phi')^i_B$ and $(\phi')^i_S$ contained ϕ provide robustness to flow discontinuity on the object boundary. In addition, $(\phi')^i_S$ is also regularizer for a gradient constraint in motion space. All of those terms can be detailed as follows:

$$
\begin{aligned}
(\phi')^i_B &= \phi'\{(b^i_z + b^i_x du^i + b^i_y dv^i)^2 \\
&\quad + \alpha(b^i_{xz} + v^i_{xx} du^i + v^i_{xy} dv^i)^2 + \alpha(b^i_{yz} + b^i_{xy} du^i + b^i_{yy} dv^i)^2\}
\end{aligned} \tag{A.23}
$$

$$
(\phi')^i_S = \phi'\{\left\|\nabla(u^i + du^i)\right\|^2 + \left\|\nabla(v^i + dv^i)\right\|^2\} \tag{A.24}
$$

Although we fixed $\mathbf{w}^i$ in Eqs. A.21 A.22, the nonlinearity in ϕ' leads to the difficulty of solving the system. The inner fixed point iterations are applied to remove this nonlinearity: $du^{i,j}$ and $dv^{i,j}$ are assumed to converge within j iterations by initializing $du^{i,0} = 0$ and $dv^{i,0} = 0$. Finally, we have the linear system in $du^{i,j+1}$ and $dv^{i,j+1}$ as follows:

$$
\begin{aligned}
&(\phi')^{i,j}_B \cdot \{b^i_x(b^i_z + b^i_x du^{i,j+1} + b^i_y dv^{i,j+1}) \\
&+\alpha b^i_{xx}(b^i_{xz} + b^i_{xx} du^{i,j+1} + b^i_{xy} dv^{i,j+1}) + \alpha b^i_{xy}(b^i_{yz} + b^i_{xy} du^{i,j+1} + b^i_{yy} dv^{i,j+1})\} \\
&-\gamma(\phi')^{i,j}_S \cdot \nabla(u^i + du^{i,j+1}) = 0
\end{aligned} \tag{A.25}
$$

$$
\begin{aligned}
&(\phi')^{i,j}_B \cdot \{b^i_y(b^i_z + b^i_x du^{i,j+1} + b^i_y dv^{i,j+1}) \\
&+\alpha b^i_{yy}(b^i_{yz} + b^i_{xy} du^{i,j+1} + b^i_{yy} dv^{i,j+1}) + \alpha b^i_{xy}(b^i_{xz} + b^i_{xx} du^{i,j+1} + b^i_{xy} dv^{i,j+1})\} \\
&-\gamma(\phi')^{i,j}_S \cdot \nabla(v^i + dv^{i,j+1}) = 0
\end{aligned} \tag{A.26}
$$

This resulting linear system in Eq (A.25,A.26) can be solved by common numerical optimisation methods such as *Gauss-Seidel* and Successive Over Relaxation (*SOR*).

The former with 45 iterations is employed in our implementations. Details for the computation of spatial gradient ∇ and $\|\nabla\|$can be found in Faisal and Barron's work [37].

A.3 RGB-NIR Optical Flow Energy optimisation

In Chapter 6, we present a novel *Invisible NIR Energy* for optical flow estimation. This provides additional information which may not be clear in the RGB channel alone. The main energy function is given as follows:

$$E(\mathbf{w}) = (1 - \lambda(\mathbf{x}))E_V(\mathbf{w}) + \lambda(\mathbf{x})E_N(\mathbf{w}) + \gamma E_S(\mathbf{w}) \tag{A.27}$$

Where $E_V(\mathbf{w})$ and $E_S(\mathbf{w})$ represent the regular intensity energy in the RGB channel ($N(\mathbf{x})$) while $E_N(\mathbf{w})$ denotes the proposed *Invisible NIR Energy*, i.e. representing the intensity in the NIR channel ($\mathrm{V}(\mathbf{x})$). $\lambda(\mathbf{x})$ is a novel *Detail-Aware Weight*, proposed as one of our other main contributions.

A.3.1 Detail-Aware Weight $\lambda(\mathbf{x})$ Initialization

As mentioned in Sec. 6.4.1, $\lambda(\mathbf{x})$ can be calculated before energy minimisation due to its independent nature against motion within a scene. We define $\lambda(\mathbf{x})$ as follows:

$$\lambda(\mathbf{x}) = \left(1 + \exp\left\{-a\left(\frac{|\Delta N_1(\mathbf{x})|}{|\Delta V_1(\mathbf{x})| + |\Delta N_1(\mathbf{x})|} - b\right)\right\}\right)^{-1} \tag{A.28}$$

Where a sigmoid function with parameters $a = 10$ and $b = 0.5$ is applied while $\Delta = (\Delta_x, \Delta_y)^T$ denotes the spacial gradient calculated using a 3×3 Sobel kernel as follows:

$$\Delta_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I \; and \; \Delta_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \tag{A.29}$$

Where I denotes image intensity w.r.t $N_1(\mathbf{x})$ or $V_1(\mathbf{x})$, while $*$ presents the 2D convolution operator. $|\Delta| = \sqrt{\Delta_x^2 + \Delta_y^2}$ is the gradient magnitude. After obtaining the weight $\lambda(\mathbf{x})$, we resize it as well as the input images to the same rate on each level of the n-level pyramid, denoted by $\lambda^i, i = 1, 2, \ldots, n$.

A.3.2 Numerical Scheme for Energy minimisation

As mentioned in Sec. 6.4.1, we follow a coarse-to-fine strategy and use nested fixed point iterations to minimise the proposed RGB-NIR energy. Note that the numerical scheme is similar to Brox *et al.* [20]. Here, we use the same abbreviations from Chapter 6:

$$
\begin{aligned}
V_x &= \partial_x V_2(\mathbf{x}+\mathbf{w}) & V_{yy} &= \partial_{yy} V_2(\mathbf{x}+\mathbf{w}) \\
V_y &= \partial_y V_2(\mathbf{x}+\mathbf{w}) & V_z &= V_2(\mathbf{x}+\mathbf{w}) - V_1(\mathbf{x}) \\
V_{xx} &= \partial_{xx} V_2(\mathbf{x}+\mathbf{w}) & V_{xz} &= \partial_x V_2(\mathbf{x}+\mathbf{w}) - \partial_x V_1(\mathbf{x}) \\
V_{xy} &= \partial_{xy} V_2(\mathbf{x}+\mathbf{w}) & V_{yz} &= \partial_y V_2(\mathbf{x}+\mathbf{w}) - \partial_y V_1(\mathbf{x}) \\
N_x &= \partial_x N_2(\mathbf{x}+\mathbf{w}) & & \\
N_y &= \partial_y N_2(\mathbf{x}+\mathbf{w}) & N_z &= N_2(\mathbf{x}+\mathbf{w}) - N_1(\mathbf{x})
\end{aligned}
$$

At the first phase of energy minimisation, Euler-Lagrange is applied to Eq. A.27:

$$
\begin{aligned}
(1-\lambda)\phi'(V_z^2 + \theta(V_{xz}^2 + V_{yz}^2)) \cdot (V_x V_z + \theta(V_{xx}V_{xz} + V_{xy}V_{yz})) \\
+\lambda\phi'(N_z^2) \cdot (N_x N_z) \\
-\gamma\phi'(\|\nabla u\|^2 + \|\nabla v\|^2) \cdot \nabla u = 0
\end{aligned} \tag{A.30}
$$

$$
\begin{aligned}
(1-\lambda)\phi'(V_z^2 + \theta(V_{xz}^2 + V_{yz}^2)) \cdot (V_y V_z + \theta(V_{yy}V_{yz} + V_{xy}V_{xz})) \\
+\lambda\phi'(N_z^2) \cdot (N_y N_z) \\
-\gamma\phi'(\|\nabla u\|^2 + \|\nabla v\|^2) \cdot \nabla v = 0
\end{aligned} \tag{A.31}
$$

Next, an n-level image pyramid is constructed from the top coarsest level to the bottom finest level. We initialize the flow field as $\mathbf{w}^0 = (0,0)^T$ on the top level and apply outer fixed point iterations on $\mathbf{w}$. We assume that the solution $\mathbf{w}^{i+1}$ converges on the $i+1$ level as:

$$
\begin{aligned}
(1-\lambda^i)\phi'((V_z^{i+1})^2 + \theta((V_{xz}^{i+1})^2 + (V_{yz}^{i+1})^2)) \cdot (V_x^i V_z^{i+1} + \theta(V_{xx}^i V_{xz}^{i+1} + V_{xy}^i V_{yz}^{i+1})) \\
+\lambda^i\phi'((N_z^{i+1})^2) \cdot N_x^i N_z^{i+1} \\
-\gamma\phi'(\left\|\nabla u^{i+1}\right\|^2 + \left\|\nabla v^{i+1}\right\|^2) \cdot \nabla u^{i+1} = 0
\end{aligned} \tag{A.32}
$$

$$
\begin{aligned}
(1-\lambda^i)\phi'((V_z^{i+1})^2 + \theta((V_{xz}^{i+1})^2 + (V_{yz}^{i+1})^2)) \cdot (V_y^i V_z^{i+1} + \theta(V_{yy}^i V_{yz}^{i+1} + V_{xy}^i V_{xz}^{i+1})) \\
+\lambda^i\phi'((N_z^{i+1})^2) \cdot N_y^i N_z^{i+1} \\
-\gamma\phi'(\left\|\nabla u^{i+1}\right\|^2 + \left\|\nabla v^{i+1}\right\|^2) \cdot \nabla v^{i+1} = 0
\end{aligned} \tag{A.33}
$$

Due to the nonlinearity of terms ϕ', V_*^{i+1} and N_z^{i+1}, the system (Eqs. A.32, A.33) is hard to solve using linear numerical methods. First order Taylor expansions are

applied to remove the nonlinearity in V_*^{i+1} and N_z^{i+1}:

$$\begin{aligned}
V_z^{i+1} &\approx V_z^i + V_x^i du^i + V_y^i dv^i \\
V_{xz}^{i+1} &\approx V_{xz}^k + V_{xx}^i du^i + V_{xy}^i dv^i \\
V_{yz}^{i+1} &\approx V_{yz}^k + V_{xy}^i du^i + V_{yy}^i dv^i \\
N_z^{i+1} &\approx N_z^i + N_x^i du^i + N_y^i dv^i
\end{aligned}$$

We follow the coarse-to-fine flow assumption of Brox *et al.* [20] w.r.t. $u^{i+1} = u^i + du^i$ and $v^{i+1} = v^i + dv^i$ where the flow field and its incremental from the current level i can be used to obtain the unknown flow field on the next level $i+1$. The new system is presented as follows:

$$\begin{aligned}
&(1-\lambda^i)(\phi')_V^i \cdot (V_x^i(V_z^i + V_x^i du^i + V_y^i dv^i) \\
+\theta\ &[V_{xx}^i(V_{xz}^i + V_{xx}^i du^i + V_{xy}^i dv^i) + V_{xy}^i(V_{yz}^i + V_{xy}^i du^i + V_{yy}^i dv^i)]) \\
&+\lambda^i(\phi')_N^i \cdot N_x^i(N_z^i + N_x^i du^i + N_y^i dv^i) \\
&-\gamma(\phi')_S^i \cdot \nabla(u^i + du^i) = 0
\end{aligned} \tag{A.34}$$

$$\begin{aligned}
&(1-\lambda^i)(\phi')_V^i \cdot (V_y^i(V_z^i + V_x^i du^i + V_y^i dv^i) \\
+\theta\ &[V_{yy}^i(V_{yz}^i + V_{xy}^i du^i + V_{yy}^i dv^i) + V_{xy}^i(V_{xz}^i + V_{xx}^i du^i + V_{xy}^i dv^i)]) \\
&+\lambda^i(\phi')_N^i \cdot N_y^i(N_z^i + N_x^i du^i + N_y^i dv^i) \\
&-\gamma(\phi')_S^i \cdot \nabla(v^i + dv^i) = 0
\end{aligned} \tag{A.35}$$

Note that the terms $(\phi')_V^i$ and $(\phi')_N^i$ contain the regularizer ϕ which provides robustness to flow discontinuity on the object boundary while $(\phi')_S^i$ is another regularizer acting as a gradient constraint in motion space. We have:

$$\begin{aligned}
(\phi')_V^i &= \phi'((V_z^i + V_x^i du^i + V_y^i dv^i)^2 \\
&\quad + \theta[(V_{xz}^i + V_{xx}^i du^i + V_{xy}^i dv^i)^2 + (V_{yz}^i + V_{xy}^i du^i + V_{yy}^i dv^i)^2]) \\
(\phi')_N^i &= \phi'(N_z^i + N_x^i du^i + N_y^i dv^i)^2 \\
(\phi')_S^i &= \phi'(\left\|\nabla(u^i + du^i)\right\|^2 + \left\|\nabla(v^i + dv^i)\right\|^2)
\end{aligned} \tag{A.36}$$

Although $\mathbf{w}^i$ is fixed in Eqs. A.34 A.35, the system is still difficult to solve because of the nonlinearity in ϕ' – in particular the du^i and dv^i. We apply an inner fixed point iteration to remove their nonlinearity: $du^{i,j}$ and $dv^{i,j}$ are proposed to converge within j iterations with the initial $du^{i,0} = 0$ and $dv^{i,0} = 0$. Therefore, we have the linear system

in $du^{i,j+1}$ and $dv^{i,j+1}$ as follows:

$$\begin{aligned}(1-\lambda^i)(\phi')^{i,j}_V \cdot (V^i_x(V^i_z + V^i_x du^{i,j+1} + V^i_y dv^{i,j+1}) \\ +\theta\ [V^i_{xx}(V^i_{xz} + V^i_{xx} du^{i,j+1} + V^i_{xy} dv^{i,j+1}) + V^i_{xy}(V^i_{yz} + V^i_{xy} du^{i,j+1} + V^i_{yy} dv^{i,j+1})]) \\ +\lambda^i(\phi')^{i,j}_N \cdot N^i_x(N^i_z + N^i_x du^{i,j+1} + N^i_y dv^{i,j+1}) \\ -\gamma(\phi')^{i,j}_S \cdot \nabla(u^i + du^{i,j+1}) = 0\end{aligned} \tag{A.37}$$

$$\begin{aligned}(1-\lambda^i)(\phi')^{i,j}_V \cdot (V^i_y(V^i_z + V^i_x du^{i,j+1} + V^i_y dv^{i,j+1}) \\ +\theta\ [V^i_{yy}(V^i_{yz} + V^i_{xy} du^{i,j+1} + V^i_{yy} dv^{i,j+1}) + V^i_{xy}(I^i_{xz} + V^i_{xx} du^{i,j+1} + V^i_{xy} dv^{i,j+1})]) \\ +\lambda^i(\phi')^{i,j}_N \cdot N^i_y(N^i_z + N^i_x du^{i,j+1} + N^i_y dv^{i,j+1}) \\ -\gamma(\phi')^{i,j}_S \cdot \nabla(v^i + dv^{i,j+1}) = 0\end{aligned} \tag{A.38}$$

This resulting linear system can be solved using common numerical optimisation methods such as *Gauss-Seidel* and Successive Over Relaxation (*SOR*). The latter is employed in our implementation. Details for the Laplacian operator ∇ can be found in Faisal and Barron's work [37].

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, *Slic superpixels compared to state-of-the-art superpixel methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'12), 34 (2012), pp. 2274–2282. 42

[2] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, *Pyramid methods in image processing*, RCA engineer, 29 (1984), pp. 33–41. 14

[3] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, *Symmetrical dense optical flow estimation with occlusions detection*, in European Conference on Computer Vision (ECCV'02), Springer, 2002, pp. 721–735. 16

[4] G. Aubert, R. Deriche, and P. Kornprobst, *Computing optical flow via variational techniques*, SIAM Journal on Applied Mathematics, 60 (1999), pp. 156–182. 10

[5] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, *A database and evaluation methodology for optical flow*, International Journal of Computer Vision (IJCV'11), 92 (2011), pp. 1–31. v, vi, 3, 22, 23, 27, 37, 42, 49, 50, 51, 53, 55, 72, 73, 75, 88, 89, 96, 98, 101

[6] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, *Performance of optical flow techniques*, International journal of computer vision (IJCV'94), 12 (1994), pp. 43–77. 22, 27

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, *Speeded-up robust features (surf)*, Computer Vision and Image Understanding (CVIU'08), 110 (2008), pp. 346–359. 17

[8] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. A. Beardsley, C. Gotsman, R. W. Sumner, and M. H. Gross, *High-quality passive facial perfor-*

mance capture using anchor frames, ACM Transactions on Graphics (TOG'11), 30 (2011), p. 75. 6, 80, 82, 84, 92

[9] M. Ben-Ezra and S. Nayar, *Motion-based motion deblurring*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'04), 26 (2004), pp. 689–698. 32

[10] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, *Hierarchical model-based motion estimation*, in European Conference on Computer Vision (ECCV'092), 1992, pp. 237–252. 14

[11] J. Bigün, G. H. Granlund, and J. Wiklund, *Multidimensional orientation estimation with applications to texture analysis and optical flow*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'91), 13 (1991), pp. 775–790. 7

[12] M. Black and P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Computer vision and image understanding (CVIU'96), 63 (1996), pp. 75–104. 7, 10, 12, 88, 90, 91, 92, 96, 109

[13] M. J. Black and P. Anandan, *Robust dynamic motion estimation over time*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'91), IEEE, 1991, pp. 296–302. 12

[14] G. Borshukov, D. Piponi, O. Larsen, J. Lewis, and C. Tempelaar-Lietz, *Universal capture: image-based facial animation for the matrix reloaded*, in ACM SIGGRAPH'05 Courses, ACM, 2005, p. 16. 79, 80

[15] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'01), 23 (2001), pp. 1222–1239. 14, 37

[16] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, *High resolution passive facial performance capture*, ACM Transactions on Graphics (TOG'10), 29 (2010), p. 41. 6, 80

[17] M. Brown and S. Susstrunk, *Multi-spectral sift for scene category recognition*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 177–184. 33, 34, 35, 95, 97

[18] T. Brox, *Smoothing of matrix-valued data*, Master Thesis in Computer Engineering, (2002). 7

[19] T. Brox, *From Pixels to Regions: Partial Differential Equations in Image Analysis*, PhD thesis, Universität des Saarlandes, 2005. 14

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High accuracy optical flow estimation based on a theory for warping*, in European Conference on Computer Vision (ECCV'04), 2004, pp. 25–36. iv, vi, 8, 9, 13, 15, 23, 37, 39, 41, 46, 48, 49, 52, 53, 54, 55, 63, 71, 72, 103, 105, 124, 126, 128, 129, 132, 134

[21] T. Brox and J. Malik, *Large displacement optical flow: Descriptor matching in variational motion estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'11), 33 (2011), pp. 500–513. 8, 9, 13, 20, 60, 79, 81, 88, 90, 91, 92, 96, 104, 109

[22] A. Bruhn, J. Weickert, and C. Schnörr, *Lucas/kanade meets horn/schunck: Combining local and global optic flow methods*, International Journal of Computer Vision (IJCV'05), 61 (2005), pp. 211–231. 15, 37

[23] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, *A naturalistic open source movie for optical flow evaluation*, in European Conference on Computer Vision (ECCV'12), 2012, pp. 611–625. 26, 27, 59, 60, 74, 96, 98

[24] X. Cao, X. Tong, Q. Dai, and S. Lin, *High resolution multispectral video capture with a hybrid camera system*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 297–304. 34, 97

[25] J. Chen, L. Yuan, C.-K. Tang, and L. Quan, *Robust dual motion deblurring*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), IEEE, 2008, pp. 1–8. 30

[26] X. Chen, J. Yang, Q. Wu, J. Zhao, and X. He, *Directional high-pass filter for blurry image analysis*, Signal Processing: Image Communication, 27 (2012), pp. 760–771. 67

[27] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, *Large displacement optical flow from nearest neighbor fields*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013, pp. 2443–2450. 16, 20

[28] S. Cho and S. Lee, *Fast motion deblurring*, ACM Transactions on Graphics (TOG'09), 28 (2009), p. 145. v, vii, 30, 31, 32, 62, 64, 66, 67, 68, 69, 73

[29] S. Cho, Y. Matsushita, and S. Lee, *Removing non-uniform motion blur from images*, in International Conference on Computer Vision (ICCV'07), IEEE, 2007, pp. 1–8. 30

[30] T. S. Cho, S. Paris, B. K. Horn, and W. T. Freeman, *Blur kernel estimation using the radon transform*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 241–248. 30

[31] D. Cosker, E. Krumhuber, and A. Hilton, *A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modelling*, in International Conference on Computer Vision (ICCV'11), 2011, pp. 2296–2303. 59

[32] T. Crivelli, P.-H. Conze, P. Robert, M. Fradet, P. Pérez, et al., *Multi-step flow fusion: towards accurate and dense correspondences in long video shots*, in British Machine Vision Conference (BMVC'12), 2012. 26

[33] P. E. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins, *A lighting reproduction approach to live-action compositing*, in ACM SIGGRAPH'02, ACM, 2002, pp. 547–556. 33, 34

[34] D. DeCarlo and D. Metaxas, *The integration of optical flow and deformable models with applications to human face shape and motion estimation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96), IEEE, 1996, pp. 231–238. 79, 80

[35] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, *Implicit fairing of irregular meshes using diffusion and curvature flow*, in ACM SIGGRAPH'99, ACM Press/Addison-Wesley Publishing Co., 1999, pp. 317–324. 29

[36] M. Drulea and S. Nedevschi, *Total variation regularization of local-global optical flow*, in Intelligent Transportation Systems (ITSC'11), IEEE, 2011, pp. 318–323. 88, 90, 91, 92, 109

[37] M. Faisal and J. Barron, *High accuracy optical flow method based on a theory for warping: Implementation and qualitative/quantitative evaluation*, Image Analysis and Recognition, (2007), pp. 513–525. 127, 131, 135

[38] G. E. Farin, *Curves and surfaces for CAGD [electronic resource]: a practical guide*, Morgan Kaufmann, 2002. 28

[39] G. Farnebäck, *Fast and accurate motion estimation using orientation tensors and parametric motion models*, in International Conference on Pattern Recognition (ICPR'00), vol. 1, IEEE, 2000, pp. 135–139. 7

[40] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, *Removing camera shake from a single photograph*, 25 (2006), pp. 787–794. 30

[41] D. Firmenichy, M. Brown, and S. Susstrunk, *Multispectral interest points for rgb-nir image registration*, in International Conference on Image Processing (ICIP'11), IEEE, 2011, pp. 181–184. 95

[42] D. J. Fleet and A. D. Jepson, *Computation of component image velocity from local phase information*, International Journal of Computer Vision (IJCV'99), 5 (1990), pp. 77–104. 27

[43] K. Frankel, S. Sousa, R. Cowan, and M. King, *Concealment of the warfighter's equipment through enhanced polymer technology*, tech. report, DTIC, 2004. v, 35

[44] C. Fredembach and S. Susstrunk, *Colouring the near-infrared*, in Color and Imaging Conference, vol. 2008, Society for Imaging Science and Technology, 2008, pp. 176–182. 33, 34, 35

[45] C. Fredembach and S. Süsstrunk, *Automatic and accurate shadow detection from (potentially) a single image using near-infrared information*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'10), 165527 (2010). 35

[46] B. Fulkerson and S. Soatto, *Really quick shift: Image segmentation on a gpu*, in Trends and Topics in Computer Vision, Springer, 2012, pp. 350–358. 83, 92

[47] R. Garg, L. Pizarro, D. Rueckert, and L. Agapito, *Dense multi-frame optic flow for non-rigid objects using subspace constraints*, in Asian Conference on Computer Vision (ACCV'10), Springer, 2010, pp. 460–473. 21

[48] R. Garg, A. Roussos, and L. Agapito, *Dense variational reconstruction of non-rigid surfaces from monocular video*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013, pp. 1272–1279. 9, 26

[49] R. Garg, A. Roussos, and L. Agapito, *A variational approach to video registration with subspace constraints*, International journal of computer vision (IJCV'13), 104 (2013), pp. 286–314. vi, vii, 9, 13, 18, 20, 22, 25, 27, 37, 38, 49, 52, 53, 54, 55, 56, 60, 80, 89, 90, 91, 96, 109

[50] A. Geiger, P. Lenz, and R. Urtasun, *Are we ready for autonomous driving? the kitti vision benchmark suite*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 3354–3361. v, 24

[51] B. Glocker, T. Heibel, N. Navab, P. Kohli, and C. Rother, *Triangleflow: Optical flow with triangulation-based higher-order likelihoods*, 2010, pp. 272–285. 9, 38

[52] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, *Non-rigid dense correspondence with applications for image enhancement*, ACM Transactions on Graphics (TOG'11), 30 (2011), p. 70. 6, 64

[53] X. He, T. Luo, S. Yuk, K. Chow, K. Wong, and R. Chung, *Motion estimation method for blurred videos and application of deblurring with spatially varying blur kernels*, in International Conference on Computer Sciences and Convergence Information Technology (ICCIT'10), 2010, pp. 355–359. 19

[54] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, *Efficient filter flow for space-variant multiframe blind deconvolution*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), 2010, pp. 607–614. 18

[55] B. Horn and B. Schunck, *Determining optical flow*, Artificial intelligence, 17 (1981), pp. 185–203. 7, 8, 9, 37, 39, 63, 88, 90, 91, 92, 96, 109

[56] B. K. P. Horn, *Robot Vision*, the MIT Press, 1986. 6

[57] J. Hoschek, D. Lasser, and L. L. Schumaker, *Fundamentals of computer aided geometric design*, AK Peters, Ltd., 1993. 28

[58] A. Humayun, O. Mac Aodha, and G. J. Brostow, *Learning to find occlusion regions*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 2161–2168. 16

[59] J. Jia, *Single image motion deblurring using transparency*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), IEEE, 2007, pp. 1–8. 30

[60] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, *Image deblurring using inertial measurement sensors*, ACM Transactions on Graphics (TOG'10), 29 (2010), p. 30. 32, 63, 64

[61] N. Joshi, R. Szeliski, and D. J. Kriegman, *Psf estimation using sharp edge prediction*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), IEEE, 2008, pp. 1–8. 30

[62] A. Kannan, B. Frey, and N. Jojic, *A generative model of dense optical flow in layers*, in Spatial Coherence for Visual Motion Analysis, Springer, 2006, pp. 104–114. 15

[63] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, *Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database*, in European Conference on Computer Vision (ECCV'12), Springer, 2012, pp. 27–40. 31

[64] V. Kolmogorov and C. Rother, *Minimizing nonsubmodular functions with graph cuts-a review*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'07), 29 (2007), pp. 1274–1279. 15

[65] D. Krishnan, T. Tay, and R. Fergus, *Blind deconvolution using a normalized sparsity measure*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 233–240. v, 30, 32

[66] D. Kundur and D. Hatzinakos, *Blind image deconvolution*, IEEE Signal Processing Magazine, 13 (1996), pp. 43–64. 29, 30

[67] L. Le Feuvre, *Modeling and deformation of surfaces defined over finite elements*, in Shape Modeling International, IEEE, 2003, pp. 175–183. 28

[68] J. Lee, V. Baines, and J. Padget, *Decoupling cognitive agents and virtual environments*, in Cognitive Agents for Virtual Environments, 2013, pp. 17–36. 65

[69] V. Lempitsky, S. Roth, and C. Rother, *Fusionflow: Discrete-continuous optimization for optical flow estimation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), IEEE, 2008, pp. 1–8. 46

[70] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman, *Motion-invariant photography*, ACM Transactions on Graphics (TOG'08), 27 (2008), p. 71. v, 32, 33, 63, 64

[71] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, *Understanding and evaluating blind deconvolution algorithms*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), IEEE, 2009, pp. 1964–1971. 30

[72] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, *Efficient marginal likelihood optimization in blind deconvolution*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 2657–2664. 30

[73] C. Li, D. Pickup, T. Saunders, D. Cosker, D. Marshall, P. Hall, and P. Willis, *Water surface modeling from a single viewpoint video.*, IEEE transactions on visualization and computer graphics (TVCG'13), 19 (2013), pp. 1242–1251. 9

[74] S. Z. Li, R. Chu, S. Liao, and L. Zhang, *Illumination invariant face recognition using near-infrared images*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'07), 29 (2007), pp. 627–639. 33

[75] W. Li, Y. Chen, J. Lee, G. Ren, and D. Cosker, *Robust optical flow estimation for continuous blurred scenes using rgb-motion imaging and directional filtering*, in IEEE Winter Conference on Application of Computer Vision (WACV'14), March 2014. 4, 78

[76] W. Li, D. Cosker, and M. Brown, *An anchor patch based optimisation framework for reducing optical flow drift in long image sequences*, in Asian Conference on Computer Vision (ACCV'12), Springer, November 2012, pp. 112–125. 4, 94

[77] W. Li, D. Cosker, M. Brown, and R. Tang, *Optical flow estimation using laplacian mesh energy*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), IEEE, June 2013, pp. 2435–2442. 4, 61

[78] Y. Li and D. P. Huttenlocher, *Learning for optical flow using stochastic optimization*, in European Conference on Computer Vision (ECCV'08), 2008, pp. 379–391. 15

[79] T. M. Lillesand, R. W. Kiefer, J. W. Chipman, et al., *Remote sensing and image interpretation.*, no. Ed. 5, John Wiley & Sons Ltd, 2004. 33

[80] C. Liu, *Beyond pixels: exploring new representations and applications for motion analysis*, PhD thesis, Massachusetts Institute of Technology, 2009. 7, 14, 72

[81] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss, *Human-assisted motion annotation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), 2008, pp. 1–8. 23

[82] C. Liu and D. Sun, *A bayesian approach to adaptive video super resolution*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 209–216. 19

[83] H. Liu, R. Chellappa, and A. Rosenfeld, *Accurate dense optical flow estimation using adaptive structure tensors and a parametric model*, IEEE Transactions on Image Processing (TIP'03), 12 (2003), pp. 1170–1180. 7

[84] D. Lowe, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision (IJCV'04), 60 (2004), pp. 91–110. 79

[85] D. Lu, E. Moran, and M. Batistella, *Linear mixture model applied to amazonian vegetation classification*, Remote Sensing of Environment, 87 (2003), pp. 456–469. 33

[86] Y. M. Lu, C. Fredembach, M. Vetterli, and S. Susstrunk, *Designing color filter arrays for the joint capture of visible and near-infrared images*, in IEEE International Conference on Image Processing (ICIP'09), IEEE, 2009, pp. 3797–3800. 33

[87] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, in International Joint Conferences on Artificial Intelligence (IJCAI'81), 1981, pp. 674–679. 7, 17, 96, 101

[88] B. McCane, K. Novins, D. Crannitch, and B. Galvin, *On benchmarking optical flow*, Computer Vision and Image Understanding (CVIU'01), 84 (2001), pp. 126–143. 23

[89] S. Meister, B. Jähne, and D. Kondermann, *Outdoor stereo camera system for the generation of real-world benchmark data sets*, Optical Engineering, 51 (2012), p. 021107. 24

[90] M. Meyer, M. Desbrun, P. Schröder, and A. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and mathematics, 3 (2002), pp. 34–57. 38, 40

[91] M. R. Meyer, S. Edwards, K. H. Hinkle, and S. E. Strom, *Near-infrared classification spectroscopy: H-band spectra of fundamental mk standards*, The Astrophysical Journal, 508 (1998), p. 397. 33

[92] H.-H. Nagel, *Constraints for the estimation of displacement vector fields from image sequences.*, in International Joint Conferences on Artificial Intelligence (IJCAI'83), 1983, pp. 945–951. 9

[93] H.-H. Nagel, *On the estimation of optical flow: Relations between different approaches and some new results*, Artificial intelligence, 33 (1987), pp. 299–324. 9

[94] H.-H. Nagel and W. Enkelmann, *An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'86), (1986), pp. 565–593. 15

[95] H.-H. Nagel and A. Gehrke, *Spatiotemporally adaptive estimation and segmentation of of-fields*, in European Conference on Computer Vision (ECCV'98), Springer, 1998, pp. 86–102. 7

[96] M. Otte and H.-H. Nagel, *Optical flow estimation: advances and comparisons*, in European Conference on Computer Vision (ECCV'94), 1994, pp. 49–60. 23, 27

[97] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, *Highly accurate optic flow computation with theoretically justified warping*, International Journal of Computer Vision (IJCV'06), 67 (2006), pp. 141–158. 15

[98] S. Park, E. Park, and H. Kim, *Image deblurring using vibration information from 3-axis accelerometer*, Journal of the Institute of Electronics Engineers of Korea. SC, System and control, 45 (2008), pp. 1–11. 33

[99] D. Pickup, C. Li, D. Cosker, P. Hall, and P. Willis, *Reconstructing mass-conserved water surfaces using shape from shading and optical flow*, in Asian Conference on Computer Vision (ACCV'10), Springer, 2010, pp. 189–201. 9

[100] D. Pizarro and A. Bartoli, *Feature-based deformable surface detection with self-occlusion reasoning*, International Journal of Computer Vision (IJCV'12), 97 (2012), pp. 54–70. iv, vi, 17, 18, 38, 49, 53, 54, 55, 83, 90, 91

[101] T. Portz, L. Zhang, and H. Jiang, *Optical flow in the presence of spatially-varying motion blur*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 1752–1759. vii, 19, 63, 72, 73, 75

[102] T. Pouli, D. W. Cunningham, and E. Reinhard, *A survey of image statistics relevant to computer graphics*, 30 (2011), pp. 1761–1788. 29

[103] S. Ricco and C. Tomasi, *Dense lagrangian motion estimation with occlusions*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 1800–1807. 16, 21

[104] S. Ricco and C. Tomasi, *Simultaneous compaction and factorization of sparse image motion matrices*, in European Conference on Computer Vision (ECCV'12), 2012, pp. 456–469. 17

[105] S. Roth and M. J. Black, *On the spatial statistics of optical flow*, International Journal of Computer Vision (IJCV'07), 74 (2007), pp. 33–50. 23

[106] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, *Optimizing binary mrfs via extended roof duality*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), IEEE, 2007, pp. 1–8. 46

[107] N. Salamati, C. Fredembach, and S. Susstrunk, *Material classification using color and nir images*, in Color and Imaging Conference, vol. 2009, Society for Imaging Science and Technology, 2009, pp. 216–222. 33

[108] M. Salzmann and P. Fua, *Reconstructing sharply folding surfaces: A convex formulation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), IEEE, 2009, pp. 1054–1061. iv, 2

[109] M. Salzmann, R. Hartley, and P. Fua, *Convex optimization for deformable surface 3-d tracking*, in International Conference on Computer Vision (ICCV'07), 2007, pp. 1–8. vi, 57, 58, 89

[110] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua, *Closed-form solution to non-rigid 3d surface registration*, in European Conference on Computer Vision (ECCV'08), Springer, 2008, pp. 581–594. vi, 57, 58

[111] L. Schaul, C. Fredembach, and S. Susstrunk, *Color image dehazing using the near-infrared*, in International Conference on Image Processing (ICIP'09), IEEE, 2009, pp. 1629–1632. 35, 95

[112] C. Schnörr, *Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class*, International Journal of Computer Vision (IJCV'91), 6 (1991), pp. 25–38. 9

[113] C. Schnörr, *On functionals with greyvalue-controlled smoothness terms for determining optical flow*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'93), 15 (1993), pp. 1074–1079. 9

[114] Y. Schoueri, M. Scaccia, and I. Rekleitis, *Optical flow from motion blurred color images*, in Canadian Conference on Computer and Robot Vision, 2009. 63

[115] P. Schröder, D. Zorin, T. DeRose, D. Forsey, L. Kobbelt, M. Lounsbery, and J. Peters, *Subdivision for modeling and animation*, ACM SIGGRAPH 2000 Course Notes, (2000). 28

[116] S. M. Seitz and S. Baker, *Filter flow*, in International Conference on Computer Vision (ICCV'09), 2009, pp. 143–150. 18

[117] A. Sellent, M. Eisemann, B. Goldlucke, D. Cremers, and M. Magnor, *Motion field estimation from alternate exposure images*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'11), 33 (2011), pp. 1577–1589. 18

[118] Q. Shan, J. Jia, and A. Agarwala, *High-quality motion deblurring from a single image*, ACM Transactions on Graphics (TOG'08), 27 (2008), p. 73. 30, 31, 66

[119] Q. Shan, W. Xiong, and J. Jia, *Rotational motion deblurring of a rigid object from a single image*, in International Conference on Computer Vision (ICCV'07), IEEE, 2007, pp. 1–8. 30

[120] D. Shulman and J.-Y. Herve, *Regularization of discontinuous flow fields*, in Workshop on Visual Motion, 1989, pp. 81–86. 10

[121] M. Snyder, *On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction*, in Workshop on Visual Motion, IEEE, 1989, pp. 107–115. 9

[122] O. Sorkine, *Laplacian mesh processing*, in State-of-the-Art Report in Eurographics, 2005, pp. 53–70. 28, 38, 40, 44

[123] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, *Laplacian surface editing*, in Eurographics/ACM SIGGRAPH symposium on Geometry processing, ACM, 2004, pp. 175–184. v, 28

[124] A. N. Stein and M. Hebert, *Occlusion boundaries from motion: low-level detection and mid-level reasoning*, International journal of computer vision (IJCV'09), 82 (2009), pp. 325–357. 16

[125] M. Stoll, S. Volz, and A. Bruhn, *Joint trilateral filtering for multiframe optical flow*. 9

[126] D. Sun, S. Roth, and M. Black, *Secrets of optical flow estimation and their principles*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), 2010, pp. 2432–2439. 8, 12, 15, 39, 81, 88, 90, 91, 92, 109

[127] D. Sun, E. B. Sudderth, and M. J. Black, *Layered image motion with explicit occlusions, temporal consistency, and depth ordering*, in Advances in Neural Information Processing Systems (NIPS'10), 2010, pp. 2226–2234. 16

[128] D. Sun, E. B. Sudderth, and M. J. Black, *Layered segmentation and optical flow estimation over time*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 1768–1775. 15, 16

[129] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, M. J. Black, D. Butler, G. Stanley, and M. Black, *A fully-connected layered model of foreground and background flow*. 16

[130] L. Sun, S. Cho, J. Wang, and J. Hays, *Edge-based blur kernel estimation using patch priors*. 30

[131] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik, *Occlusion boundary detection and figure/ground assignment from optical flow*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011, pp. 2233–2240. 16

[132] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer-Verlag New York, Inc., 2010. 7, 20

[133] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin, *Image/video deblurring using a hybrid camera*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), 2008, pp. 1–8. 32, 62

[134] Y.-W. Tai and S. Lin, *Motion-aware noise filtering for deblurring of noisy and blurry images*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), 2012, pp. 17–24. 66

[135] R. Tang, D. Cosker, and W. Li, *Global alignment for dynamic 3d morphable model construction*, in Workshop on Vision and Language (V&LW'12), 2012. 5

[136] M. Tistarelli, *Multiple constraints for optical flow*, in European Conference on Computer Vision (ECCV'94), Springer, 1994, pp. 61–70. 8

[137] C. Tomasi and T. Kanade, *Detection and tracking of point features*, in Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991. 17

[138] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler, *Tracking and modeling non-rigid objects with rank constraints*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), IEEE, 2001, pp. 493–500. 21

[139] W. Trobin, T. Pock, D. Cremers, and H. Bischof, *Continuous energy minimization via repeated binary fusion*, European Conference on Computer Vision (ECCV'08), (2008), pp. 677–690. 37

[140] C. J. Tucker, *Remote sensing of leaf water content in the near infrared*, Remote Sensing of Environment, 10 (1980), pp. 23–32. 33

[141] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, *Modeling temporal coherence for optical flow*, in International Conference on Computer Vision (ICCV'11), 2011, pp. 1116–1123. iv, 9, 10, 21

[142] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, *Discrete quadratic curvature energies*, Computer Aided Geometric Design, 24 (2007), pp. 499–518. 40

[143] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, *An improved algorithm for tv-l 1 optical flow*, in Statistical and Geometrical Approaches to Visual Motion Analysis, Springer, 2009, pp. 23–45. vi, 11, 15, 49, 52, 53, 54, 55, 82, 88, 90, 91, 92, 109

[144] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, *DeepFlow: Large displacement optical flow with deep matching*, in IEEE Intenational Conference on Computer Vision (ICCV'13), Dec. 2013. 20

[145] W. Welch and A. Witkin, *Free-form shape design using triangulated surfaces*, in ACM SIGGRAPH'94, ACM, 1994, pp. 247–256. 28

[146] M. Werlberger, T. Pock, and H. Bischof, *Motion estimation with non-local total variation regularization*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), IEEE, 2010, pp. 2464–2471. 11

[147] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, *Anisotropic huber-l1 optical flow.*, in British Machine Vision Conference (BMVC'09), vol. 1, 2009, p. 3. 13

[148] R. White, K. Crane, and D. Forsyth, *Capturing and animating occluded cloth*, ACM Transactions on Graphics (TOG'07), 26 (2007), p. 34. 52, 89

[149] P. Williams and K. Norris, *Near-infrared technology in the agricultural and food industries*, American Association of Cereal Chemists, Inc., 1987. 33

[150] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, *Lessons and insights from creating a synthetic optical flow benchmark*, in ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation (ECCVW'12), 2012, pp. 168–177. 59, 74

[151] L. Xu and J. Jia, *Two-phase kernel estimation for robust motion deblurring*, in European Conference on Computer Vision (ECCV'10), 2010, pp. 157–170. v, 30, 31, 32

[152] L. Xu, J. Jia, and Y. Matsushita, *Motion detail preserving optical flow estimation*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), 2010, pp. 1293–1300. 8, 14, 15, 16, 45, 46, 60, 96, 109

[153] L. Xu, S. Zheng, and J. Jia, *Unnatural l0 sparse representation for natural image deblurring*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013, pp. 1107–1114. 63, 66, 72

[154] Y. Yacoob and L. S. Davis, *Temporal multi-scale models for flow and acceleration*, International Journal of Computer Vision (IJCV'99), 32 (1999), pp. 147–163. 7

[155] R. Yang, Z. Wang, S. Liu, and X. Wu, *Design of an accurate near infrared optical tracking system in surgical navigation*, Journal of Lightwave Technology, 31 (2013), pp. 223–231. 35

[156] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, *Progressive inter-scale and intra-scale non-blind image deconvolution*, 27 (2008), p. 74. 63

[157] C. Zach, T. Pock, and H. Bischof, *A duality based approach for realtime tv-l 1 optical flow*, 4713 (2007), pp. 214–223. 11

[158] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang, *Handling noise in single image deblurring using directional filters*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013, pp. 612–619. v, 31, 62, 66, 70

[159] H. Zimmer, A. Bruhn, and J. Weickert, *Optic flow in harmony*, International Journal of Computer Vision (IJCV'11), 93 (2011), pp. 368–388. 15, 21, 37

[160] C. Zitnick, N. Jojic, and S. B. Kang, *Consistent segmentation for optical flow estimation*, in International Conference on Computer Vision (ICCV'05), 2005, pp. 1308–1315. 6